T.C.

ISTANBUL SABAHATTIN ZAIM UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

PROGRAM OF COMPUTER SCIENCE AND ENGINEERING

IMPROVEMENT OF ENERGY EFFICIENT IN LOW POWER WIRELESS SENSOR NETWORKS

Ph.D. DISSERTATION

Amir SEYYEDABBASI

Istanbul

January, 2020

T.C.

ISTANBUL SABAHATTIN ZAIM UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

PROGRAM OF COMPUTER SCIENCE AND ENGINEERING

IMPROVEMENT OF ENERGY EFFICIENT IN LOW POWER WIRELESS SENSOR NETWORKS

Ph.D. DISSERTATION

Amir SEYYEDABBASI

Thesis Advisor

Asst. Prof.Dr. Farzad KIANI

Istanbul

January, 2020

This study has been approved in partial fulfillment of the requirements for Ph.D. Degree in Computer Science And Engineering.

Chair person of jury Dr. Öğr. Üyesi Farzad KIANI

Member of jury Prof. Dr. ZAFER ASLAN

Member of jury Dr. Öğr. Üyesi Aydın Tarık ZENGİN

Member of jury Dr. Öğr. Üyesi Gökhan ERDEMİR

Member of jury Dr. Öğr. Üyesi Ali HAMİTOĞLU

Approval by Institute of Sciences

Prof. Dr. Ahmet Korhan BİNARK

Director,

Instititute of Sciences

DECLARATION OF SCIENTIFIC ETHICS AND ORIGINALITY

This is to certify that this Ph.D thesis dissertation titled "**Improvement of Energy Efficient in Low Power Wireless Sensor Networks** " is my own work and I have acted according to scientific ethics and academic rules while producing it. I have collected and used all information and data according to scientific ethics and guidelines on thesis writing of Sabahattin Zaim University. I have fully referenced, in both the text and bibliography, all direct and indirect quotations and all sources I have used in this work.

Amir SEYYEDABBASI

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my advisor Asst. Prof. Dr. Farzad KIANI for the continuous support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. Besides my advisor, I would like to thank the rest of my thesis committee: Asst. Prof. Dr. Aydin Tarik ZENGIN and Asst. Prof. Dr. Gokhan ERDEMIR. My sincere thanks also to Assoc. Prof. Dr. Gulustan DOGAN who gave kind advice throughout my Ph.D. research studies. Without they precious support it would not be possible to conduct this research.

I thank my fellow lab mates we were working together, and for all the fun, we have had in the last four years. In addition, I wish to thank my best friends, for their love, care, and moral support.

Last but not the least, I would like to thank my family: my parents and to my brothers and sister for supporting me spiritually throughout writing this thesis and my life in general.

Yours truly Amir SEYYEDABBASI

Istanbul – 2020

ABSTRACT

Improvement of Energy Efficient in Low Power Wireless Sensor Networks

Amir SEYYEDABBASI

Ph.D., Computer Science and Engineering

Supervisor: Asst. Prof. Dr. Farzad KIANI

January 2020, 176 + xvi Pages

Wireless Sensor Networks (WSN) consists of many sensors that have connectivity with each other and run cooperatively, these sensor nodes can send and receive data from in the deployment area. WSN in the last decade's one of the main research concepts between researchers. With the spread of technology and the internet, wireless sensor networks are becoming increasingly visible in everyday life. Sensor nodes have some constraints, for instance, sensor nodes have limited memory, battery, range. Today, there is much research to solve some constraints, on other hand, sensor nodes have some advantages like easy to deploy, some mobility of nodes, ability to cope with node failures (resilience), ease of use and ability to withstand harsh environmental conditions. In this thesis, we have focused on improving the energy consumption of wireless sensor networks. This concept actually is related to routing algorithms, data aggregation, and deployment in wireless sensor networks. Many algorithms present energy efficiency in wireless sensor networks. The proposed methods in this thesis, outlines in bellow:

The first proposed method of this study is a new method for clustering. The clustering methods increase energy consumption. The protocol considered four parameters to select optimal cluster head (HEEL) such as; node's energy, node's neighbors' energy, hop size to the base station, node links to neighbors to achieve energy efficiency.

The second method proposed cluster selection in heterogeneous sensor networks with making the virtual grid to select twice the cluster head (EEHRSN). In this type of network, there are different types of sensor nodes. In the EEHRSN, there are three types of sensor nodes such as; normal node, advanced node, and super node. These types of the sensor have a difference in energy level and transmission range.

The third method improved the HEEL algorithm by two metaheuristic algorithms. The metaheuristic algorithms are I-GWO and Ex-GWO. Improved HEEL algorithm (I-HEEL) has two steps; the first step updates coefficients by I-GWO and Ex-GWO. The second step selects the optimal cluster heads. The protocol's main major has reached energy efficiency.

The fourth method proposed a new routing algorithm by metaheuristic algorithms. This method is an energy-efficient routing algorithm applied by two metaheuristic algorithms (I-GWO and Ex-GWO). The algorithm focused on each sensor node features like energy, distance, hop size to find optimal path between source and destination.

The last method proposed modified ant colony optimization (ACO) a new energy-efficient method in the routing algorithm for WSNs. This protocol tries to transfer aggregated data to the base station with the goal of reducing energy consumption. In this protocol, the modified version of the ACO proposed too. The simulation results on the proposed method have claimed that the resources consumed efficiency in the network.

All the proposed methods in this study simulated by MATLAB. In addition, the input configuration is the same for all comparison algorithms too. All algorithms evaluated by metric parameters such as residual energy, network lifetime, and throughput.

Keywords: Wireless sensor network, routing algorithms, clustering methods, energy efficiency, cluster head selection.

ÖZET

Düşük Güçlü Kablosuz Sensör Ağlarında Enerji Verimliliğinin Artırılması

Amir SEYYEDABBASI

Doktora, Bilgisayar Bilimi ve Mühendisliği

Tez danışmanı: Dr. Öğr. Üyesi Farzad KIANI

Ocak 2020, 176 + xvi Sayfa

Kablosuz Algılayıcı Ağlar (KAA), birbirleriyle bağlantısı olan ve işbirliği içinde çalışan birçok sensörden oluşur, bu sensör düğümleri dağıtım alanından veri gönderip alabilir. Son on yılarda KAA, araştırmacılar arasındaki temel araştırma kavramlarından biridir. Teknolojinin ve internetin yayılmasıyla birlikte, kablosuz algılayıcı ağlar günlük yaşamda giderek daha görünür hale geliyor. Algılayıcı düğümlerinin bazı kısıtlamaları vardır, örneğin, algılayıcı düğümleri sınırlı hafızaya, bataryaya, aralığa sahiptir. Günümüzde bazı kısıtlamaları çözecek çok araştırma var, diğer taraftan, sensör düğümlerinin dağitimi kolay, düğümlerin hareketliliği, düğüm sorunlarıyla başa çıkma yeteneği (esneklik), kullanım kolaylığı ve sert çevresel koşullara dayanma yeteneği gibi bazı avantajları var. Bu tezde, kablosuz algılayıcı ağların enerji tüketimini azaltmaya odaklandık. Bu konsept aslında yönlendirme algoritmaları, veri toplama ve kablosuz sensör ağlarında dağıtım ile ilgilidir. Kablosuz algılayıcı ağlarında enerji verimliliği sağlayan birçok algoritma vardır. Bu tezde önerilen yöntemler, aşağıdaki gibi belirtmektedir:

Bu çalışmanın ilk önerilen yöntemi kümeleme için yeni bir yöntemdir. Kümeleme yöntemleri enerji tüketimini arttırır. Protokol, aşağıdaki gibi optimum küme başı (HEEL) seçmek için dört parametre olarak önerildi; düğümün enerjisi, düğümün enerjisi, baz istasyonuna atlama büyüklüğü, komşu sayısı enerji verimliliği sağlamak için kullanılmaktadır.

İkinci yöntem, sanal kümelerle iki kez küme başını seçmesini sağlayan heterojen algılayıcı ağlarında önerdildi (EEHRSN). Bu ağda, farklı tipte sensör düğümleri vardır. EEHRSN'de üç tip algılayıcı düğümü vardır; normal düğüm, gelişmiş düğüm ve süper düğüm. Bu düğümtipleri enerji seviyesi ve aktarım aralığında farklılık gösterir.

Üçüncü yöntem, HEEL algoritmasını üç metaheuristik algoritma ile geliştirmiştir. Metaheuristik algoritmalar GWO, I-GWO ve Ex-GWO'dur. Geliştirilmiş HEEL algoritması (I-HEEL) iki basamağa sahiptir, ilk adım GWO, I-GWO ve Ex-GWO tarafından katsayıları günceller. İkinci adım, optimum küme kafalarını seçer. Protokolün asıl amacı enerji verimliliğine ulaşmaktır.

Dördüncü yöntem metaheuristik algoritmalar tarafından yeni bir yönlendirme algoritması önermiştir. Bu yöntem, iki metaheuristik algoritma (I-GWO ve Ex-GWO) tarafından uygulanan enerji verimli bir yönlendirme algoritmasıdır. Her algılayıcı düğümüne odaklanan algoritma, kaynak ve hedef arasında en uygun yolu bulmak için enerji, mesafe, atlama boyutu gibi özelliklere sahiptir.

Son yöntem, KAA'lar için yönlendirme algoritmasında yeni bir enerji verimli yöntem olan modifiye karınca kolonisi optimizasyonunu (KKO) önermiştir. Bu protokol, enerji tüketimini azaltmak amacıyla toplanmış verileri baz istasyonuna aktarmaya çalışır. Bu protokolde, KKO'nun değiştirilmiş versiyonu da önerilmiştir. Önerilen yönteme ilişkin simülasyon sonuçları, kaynakların ağda verimliliği tükettiğini ıspat etmiştir.

Bu çalışmada önerilen tüm yöntemler MATLAB’de simüle edilmiştir. Ayrıca, giriş konfigürasyonu tüm karşılaştırma algoritmaları için de aynıdır. Tüm algoritmalar, kalan enerji, ağ ömrü ve verimlilik gibi metrik parametrelerle değerlendirilmiştir.

Anahtar sözcükler: Kablosuz sensör ağı, yönlendirme algoritmaları, kümeleme metotları, enerji verimliliği, küme başı seçimi.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE: INTRODUCTION

1.1.Wireless Sensor Networks

A swarm of sensor nodes that could sense the environment and work cooperatively makes a wireless sensor network (WSN) (Akyildiz et. al., 2010). Advances in the micro-electro-mechanical system (MEMS) and sensor nodes sets become popular by growing telecommunication technology. Furthermore, sensor nodes are multifunctional, low-cost and easy to assemble. Sometimes sensor nodes names as motes (Deshpande et. al., 2004). WSN is one type of Ad-hoc network that does not needs access points. Sensor nodes are able to sense the environment, data processing and communication that work together consists of a wireless sensor network. WSN is one of the inseparable parts of human life. In addition, WSN is the base of the Internet of Things (IoT). Wireless sensor networks have focused on many academic researchers in recent decades.

WSN has different application scenarios of use such as; health, industry, agriculture, military, traffic, weather. For instance, with WSN, the end-user can monitor the farmland features like humidity, moisture. Also, the nurse from the health station can follow the patient blood tension, heart pulse. Sensor nodes are deployed in an area to detect pheromone or unexpected events like fire, avalanche and the volcano. Sensor node besides transmission the raw data can process the data to carry out to other sensor nodes or the sink/base station (BS) (see Figure 1.1). The sink/BS is connected to the internet and serves as a gateway to forward data to the server. In some cases, the sink/BS acts as coordinator. The transmission of the sensor nodes needs communication protocols. In the later sections, the characteristic of the sensor node and communication protocols will be explained.

The researchers in the WSN have focused on some trendy topics like; deployment, target tracking, routing algorithms, security, quality of service (QoS), MAC protocols and topology control. Similarly, energy efficiency is one of the hot topics in WSN. Energy efficiency accomplishes by reducing computation and transmission. There are different ways to approach this goal in a manner like data aggregation, clustering method, routing algorithms.

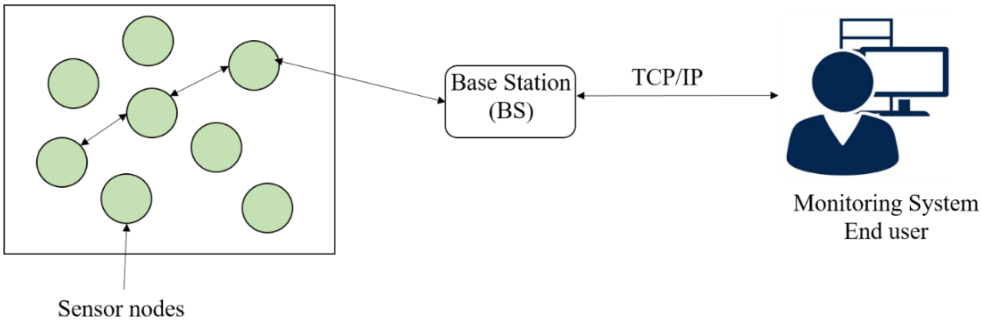


Figure 1. 1 WSN architecture

1.2.Characteristics of wireless sensor networks

Generally, due to the low cost of sensors, wireless sensor networks based on a number of a high density of sensor nodes. Sensor nodes have a tiny amount of CPU and memory. Usually, sensor nodes equipped by multifunctional embedded sensing elements. The hardware of each sensor node compounded by four components (see Figure 1.2). A sending unit that equipped by multiple sensor types such as; humidity, moisture, temperature, stabilizer track. The sensing unit consists of sensors and analog-to-digital converters. The processing unit with limited CPU serves as the processor of each sensor node. The sensed data first processed in the processing unit after that processed data store in the memory of each sensor node. The memory of the sensor node is programmable and based on application. Also, the storage of sensor nodes is limited and it is one of the challenges in WSN. The transmission unit is responsible for the transfer of processed data to other sensor nodes or BS. Antennas of sensor nodes are located in this unit. The other unit of each sensor node is the power unit. The power unit is one of the main parts of each mote. Each sensor node has a battery with limited energy. Also, the network lifetime is directly related to this unit. Usually, the power supply cannot charge along the network lifetime, because most WSN located in a harsh environment. Although new technologies like solar cells can solve this problem most researchers try to design algorithms to improve battery consumption in the application layer of WSN (Akyildiz et. al., 2002). These components are the main unit of each sensor node but other elements can be included like the GPS module, power generator, a control actuator, and other application-depend elements (Sohraby et. al., 2007).

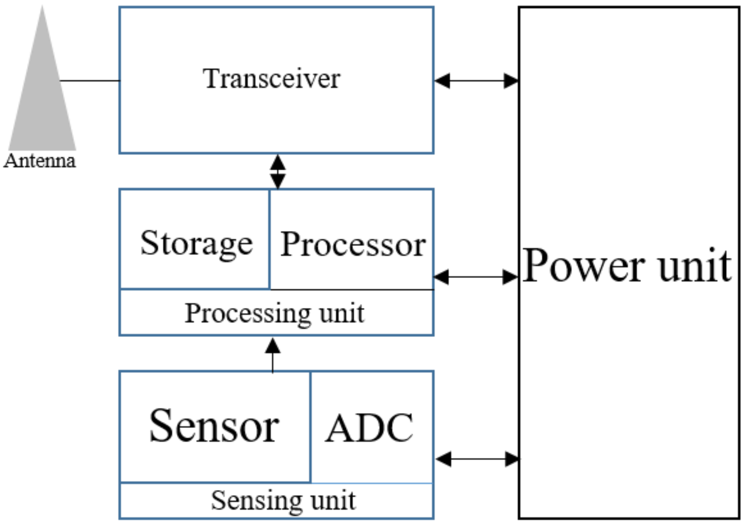


Figure 1. 2 Typical sensor node

Sensor nodes have advantages like easy to deploy, some mobility of nodes, ability to cope with node failures (resilience), ease of use and ability to withstand harsh environmental conditions. The most challenges of sensor nodes are the limited power supply and processor. Sensor nodes with short distance and single or multi-hop transmission interwork in a network. The neighbor nodes aggregate the data and transmit them to the upstream neighbor. They usually collect data from neighbors and forward aggregated data to sink/BS. Server and end-user reach the aggregated data via TCP/IP protocol that connected with sink/BS. As mentioned before in WSN sensor nodes deployed in the high-density schemes and generally deployed randomly, but the deployment scheme is based on application.

That is to say, the standardization of the sensor network is necessary; the IEEE 802.15.4 (Akyildiz et. al., 2010; Sohraby et. al., 2007) standard is defined in wireless technology. There are different bands for communication in this standard. 2.4 GHz is global, 915 MHz is used in the Americas and 868 MHz is accepted in Europe. MAC layers (Medium Access Control) layer provides for the star, mesh and hierarchical topologies communications (Akyildiz et. al., 2010).

Most of the researches focused on homogeneous sensor nodes. Also, there is a network with heterogeneous sensor nodes. In heterogeneous WSN, there are different sensor node types that there are differences in power supply, transmission range and even in the processor unit (Ilyas et. al., 2005). Heterogeneous WSN is that sensor nodes are not the same hardware design. IoT is one of the main topics by increasing IT technologies. In most IoT system, there are various sensor nodes. Therefore, IoT systems are one example of heterogeneous WSN (Figure 1.3).

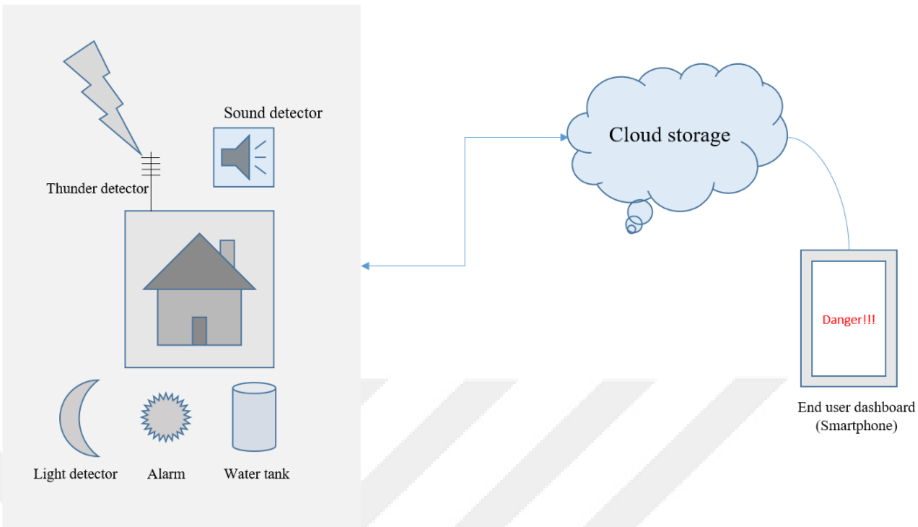


Figure 1. 3 Smart home

In IoT, sensor nodes acts as a thing. In this type of network, there are different types of motes. So, they have a difference in most of sensor node features. As seen in Figure 1.3 there are five types of sensors. The duty of motes also different in this example, for instance, the water tank has a sensor that sends data to base statin when the water reaches a minimum threshold value. On the other hand, the thunder detector sensor always must be awake to send a warning message to the electricity sensor to disconnect the electricity in a dangerous time. After all, the functionality of this sensor is different and needs a specific routing algorithm, data dissemination, and communication protocol. ZigBee standard allows sensor nodes to have different duty-cycle.

In WSN, each sensor has two different functionality source functions and router functions. When a sensor node has source function, it only transmits their data to the sink/BS. In the router function sensor node in addition to transmitting its data, it also acts as a neighbor node transmitter and participates in multi-hop communication. It is also based on application and the network topology. In mesh network topology, each device can be a source node or a router. Mesh network topology is suitable in WSN.

1.3. Routing protocols in Wireless Sensor Networks

Routing is one of the key principles in WSN. There are many challenges in designing routing protocols. The main goal of routing protocols is to use sensor nodes resources efficient way due to limited energy supply. However, some typical routing protocols may not be suitable because of routing metrics. Routing protocols have the main rule to improve network lifetime. As the routing protocols helps to improve network lifetime instead of changing sensor nodes battery. Also, routing protocols dictate the role of each sensor node in the network lifetime. In WSN there are thousands of motes that have different functionality. Routing protocol dictates each mote to act as a router, coordinator or end device.

Also, routing protocols designed to increase reliability, robustness and decrease energy consumption, redundancy. In routing protocols must balance between the aggregated data and energy, due to redundancy of collected data will be happened. Also, delay, duplicate data is a reason for decreasing network lifetime. As mentioned before, the WSN consisted of thousands of sensor nodes, so the routing protocols must able to handle large-scale networks. Furthermore, the sensor nodes are the low-cost technology, the routing protocols help to use these nodes in an optimal way.

Most of the researchers focused on designing novel routing protocols because the hardware modifying is the WSN that is expensive. The popular advantage of WSN is its low cost. Solar energy utilization in wireless sensor networks is expensive. Since the sensor nodes have some limitations in resources. The routing protocols uses of the resources efficiently. In this thesis, we try to prolong the network lifetime by designing novel methods of routing protocols and topology control.

The routing protocols are one of the methods to improve network lifetime. Generally, routing protocols are divided into three categories. Flat network architecture, hierarchical networks and location-based networks (Sohraby et. al., 2007). Figure 1.4 shows the classification of routing protocols. In this thesis, we focus on the hierarchical network. This type of routing protocol based on cluster. As such, the concept of hierarchical routing is also utilized to perform energy-efficient routing in WSNs.

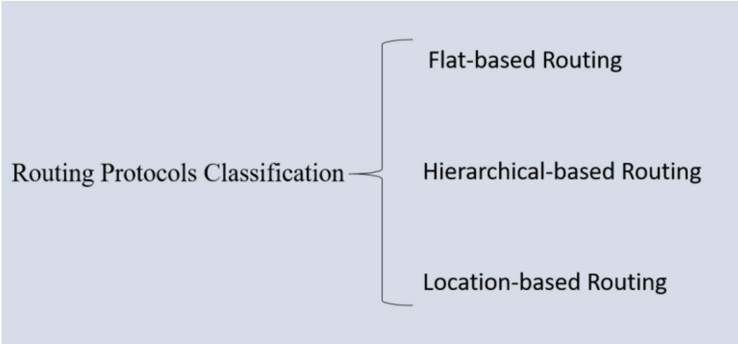


Figure 1. 4 Routing protocols classification

Some major reasons consume energy in routing protocols. Most of the routing protocols need knowledge of each sensor node location to exchange information. So, the routing protocols try to discover neighborhoods. In this way, sensor nodes consume energy for discovering. In this case, the overhead of the protocol increases. Therefore, minimizing local information exchange improves energy efficiency in routing protocol. Communication and computation cost is also important in routing protocols. In point of view energy consumption, computation is much cheaper than communication in WSN (Akyildiz et. al., 2010). In the routing, much computation is better than more data transmission in the network. The computation on each sensor can reduce data transmission times.

In WSN there is a large amount of information because many sensor nodes are deployed in a high-density area. Routing protocols must operate this information in an efficient way to reduce energy consumption. Actually, the individual information of each sensor not important than transferring packet data. So, the routing protocol should handle this knowledge in a large number of sensor nodes. Also, the routing protocol should be scalable to thousands of sensor nodes.

The sensor nodes that consisted of a WSN are low-cost. As a result, in some condition states in unexpected failures such as drain battery, crush by hardware problems or something like this sensor node may be lost the packet. However, it has a rule in a network to transfer data packet. Routing protocol should ensure robustness to node failures and prevent a single point of failure situation (Akyildiz et. al., 2010). Also, the routing protocol should operates the data packet delivery in unexpected failures even in a harsh environment.

As almost of WSN topology is supposed to be static, the routing protocols should be adaptive to network topology. In most WSN the deployment of sensor nodes is a randomly or predetermined strategy. Usually, in WSN some senor nodes maybe not transmission range to transfer data packets to sink/BS. In this way, the routing protocols should provide a path to deliver the data packet to the server (Boukergche et. al., 2008). In some wireless networks, the sensor nodes have mobility and not static. Furthermore, the routing protocol should be aware of the sensor nodes transfer circuitry. In some networks, the sleep states of nodes changed dynamically. As a result, the routing protocols should moderate with sudden changes in the wireless sensor network. Moreover, the other important clues in routing protocols are the application of the WSN. In traffic monitoring application the routing protocol varies from the network that monitoring the farmland moisture. Because of the agriculture monitoring network the sensor node, most of the time is on sleep mode. As a result, the communication between sensor nodes and sink/BS in a periodic manner. On the other hand, in the traffic control network, the sensor nodes should be communicated with each other in each time to ensure deliver traffic status to control system manager. Overall, the routing protocols should be providing these:

- Fault tolerance
- Packet delivery
- Robustness
- Scalability
- Minimal overhead

1.3.1. Flat-based protocols

In this category of routing protocols each node plays, the same role and similar in all respects. Sensor nodes collaborate to perform the sensing task. This kind of protocol is data-centric routing, that the base station broadcast queries sensor nodes and wait until they get a reply from them. As mentioned before, in WSN there are many sensor nodes in this type of protocol they have not specific identification number (ID). There is some routing algorithm in flat protocols such as; flooding, gossiping, spin, and directed diffusion. These protocols motivated the design of many other protocols.

1.3.1.1.Flooding and Gossiping

Flooding and gossiping are the common routing algorithms developed for wired and wireless networks (Sohraby et. al., 2007). In flooding algorithm sensor nodes broadcasts the received packet to all neighbors. The broadcast mechanism continues until all sensor nodes in the network received that packet. The flooding algorithm is suitable for networks that neighbors' information is not important. It's also a low-cost routing algorithm because there is not a rule to discover a complex and new path. Flooding also has some disadvantages such as; implosion, overlap and resource blindness. Implosion occurred in the flooding topology network because there is not any rule to avoid broadcast packets (Figure 1.5). Duplicate messages also broadcast from the same node. Also, two neighbor nodes may cover near regions (Figure 1.6). So two nodes sense the same region and the neighbor's node receives duplicate message. Furthermore, the flooding algorithm does not has a mechanism for energy consumption efficiently.

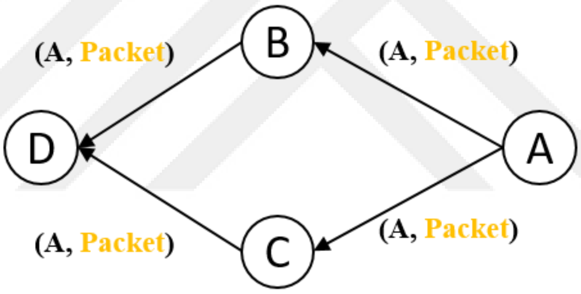


Figure 1. 5 Flooding implosion problem

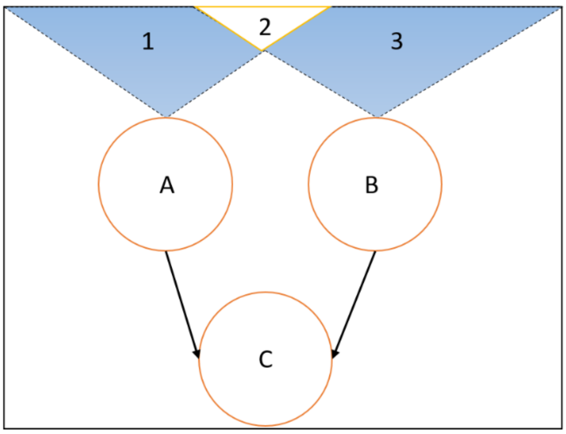


Figure 1. 6 Flooding overlap problem

The gossiping algorithm avoids some deficiencies of the flooding algorithm. Implosion solved by selecting a single node to transmission. The gossiping algorithm provides a random selection to packet delivery from neighbors, also this transmission continues in the whole network. Furthermore, the gossiping algorithm sends one copy of each message for any node but this leads to latency in the network. Generally, the energy consumption of the gossiping algorithm is lower than the flooding algorithm by avoiding the transfer duplicate message. After all, these two algorithms can be used in the network, for instance in the deployment phase or when sink/BS sends a query to collect information from the network.

1.3.1.2.Sensor Protocols for Information via Negotiation (SPIN)

Sensor Protocols for Information via Negotiation (Kulik et. al., 2002): The SPIN algorithm is one of the flat-based networks. The main objective of SPIN is to use optimal resources and defect the flooding protocols. In the classical flooding mechanism, each node sends a list of the data to the neighbors, in this mechanism the probability of losing data is very high. Also, there is not any data delivery check, so the redundancy is high too because the sensor node always sends data to the neighbors. This leads to the network implosion problem. Furthermore, network bandwidth is also overused due to high data transmission and the use of the resources in blindness way.

SPIN has two main steps, first sensor nodes negotiate with each other instead of sending all data. Second is each sensor should be aware of its energy resources to perform decisions that named resource-adaption (Kulik et. al., 2002). This algorithm is based on negotiation, every node use meta-data between each other to send and receive data. Data exchange is done by three types of messages; advertisement (ADV), request (REQ), DATA (Figure 1.7). Each node wants to exchange data with neighbors broadcast the ADV message (Step 1). Sensor nodes that receives ADV messages open the meta-data and read it, if a node is interested, replies by REQ message (Step 2). After this at the end step, the DATA packet sent to the node that requests it (Step 3).

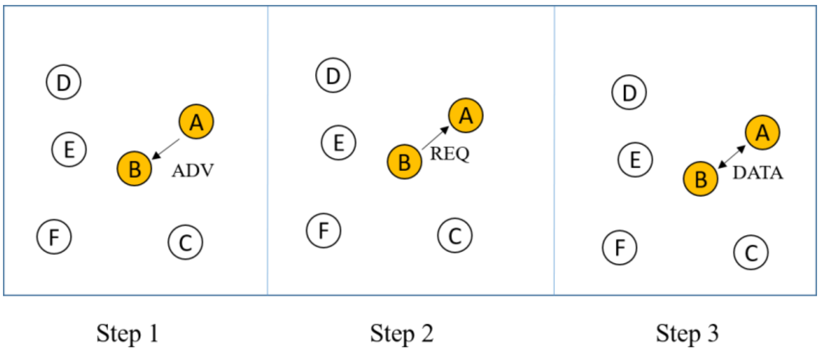


Figure 1. 7 The SPIN protocol two nodes negotiation

SPIN algorithm negotiation mechanism also applied in multiple nodes (Figure 1.8). Node B sends the ADV message to neighbors. Interested nodes reply back with REQ message, in this situation, some of the nodes may be interested, so answer by REQ message. In this example, nodes E, F answered. As a result node B, transfer DATA only with E, F nodes. The SPIN algorithm also designed for mobile networks. By changes in the local topology of the network, the SPIN algorithm adaptive with mobile sensor nodes (Kulik et. al., 2002; Martorosyan et. al., 2008). Sensor nodes in this protocol consume little energy in computation because routing in SPIN is single-hop.

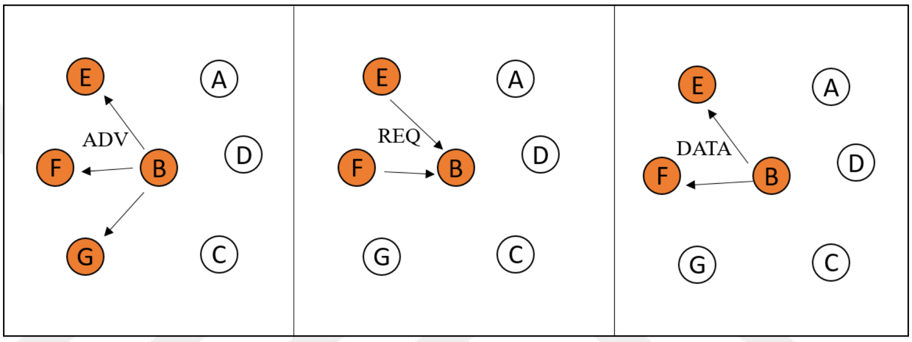


Figure 1. 8 The SPIN protocol with multiple nodes negotiation

Also (Kulik et. al., 2002) proposed four protocols based on the SPIN algorithm. Table 1.1 describes these protocol's goals.

Table 1.1 The SPIN protocol variants

Protocol name	Description
SPIN-PP	• Networks using point-to-point transmission data • A three-stage handshake protocol • Receiver node checks to see whether it has already received or not • Nodes should not respond to each message • Nodes aggregate data packets • Designed for the network with symmetric communication links

SPIN-BC	• improved version of SPIN-PP by one-to-many broadcasting • Suitable for broadcast media • Communication in the protocols by a single shared channel • Each node within a certain range receive the sender data packet • sending answer occurred when the channel is idle • one-to-many communication in this protocol
SPIN-EC	• This is a version of SPIN-PP with a low-energy threshold • Neighbors are aware of each other energy level • Transfers adaptive by neighbors energy level
SPIN-RL	• Improved version of SPIN-BC for lossy networks • improve the reliability in comparison of SPIN-BC • nodes in this protocol track of getting ADV message from which destination • resend data limited in this protocol • wait a predetermined of time to get a response from receiver then responded to any more requests

This protocol has a problem in negotiation due to maybe some unsafe nodes available in the network and gets whole data of the network. However, energy consumption is successful in comparison to the classic flooding mechanism. Also, the SPIN algorithm does not guarantee data delivery over regular intervals.

1.3.1.3.Directed Diffusion (DD)

This algorithm is a type of flat-based algorithms that relies on flooding some tasks (Intanagonwiwat et. al., 2000). DD is on-demand protocols. It means that the base station broadcast queries to the network by flooding. DD extends the network lifetime and reduces the number of transmissions; DD supports reliability and data aggregation (Jamal et. al., 2004). There is the main difference between DD and SPIN that it is in the SPIN algorithm sensor nodes advertise the availability of data while in DD it based on demand.

As mentioned before, the basic of DD is on-demand methods. DD algorithm starts transmission with interested nodes. The goal of DD is to find an efficient path between transmission and base station. In this protocol, each node remembers the node that receives the information from it and creates a gradient for the node. Four stages construct routes between the interested nodes and sink/BS.

1) Interest propagation
2) Gradient setup
3) Reinforcement
4) Data delivery

In the DD algorithm, a task announced by the packet to data matching. As mentioned that there are four stages in DD, the first step is an interest propagation. It is defined using a list of attribute-value pairs such as; interval, type, geographical location, duration etc. The interest message initiated by the sink/BS (Figure 1.9). In this stage, each sensor node stores it in an interesting cache. The duration field is used for storing the message at a specific time. The interested node forwards the message to its downstream neighbors by flooding. In some networks, maybe there are some rules to forwarding strategy.

After this stage, each node that received the interest message creates the gradient. The gradient is a response link with the neighbor and the node that received the interest. The gradient has some fields like data rate, duration and expiration time. There is any limit the number of gradients creates for a node, also between the source and the same interest. Generally, the data can be sent via various paths to the sink/BS.

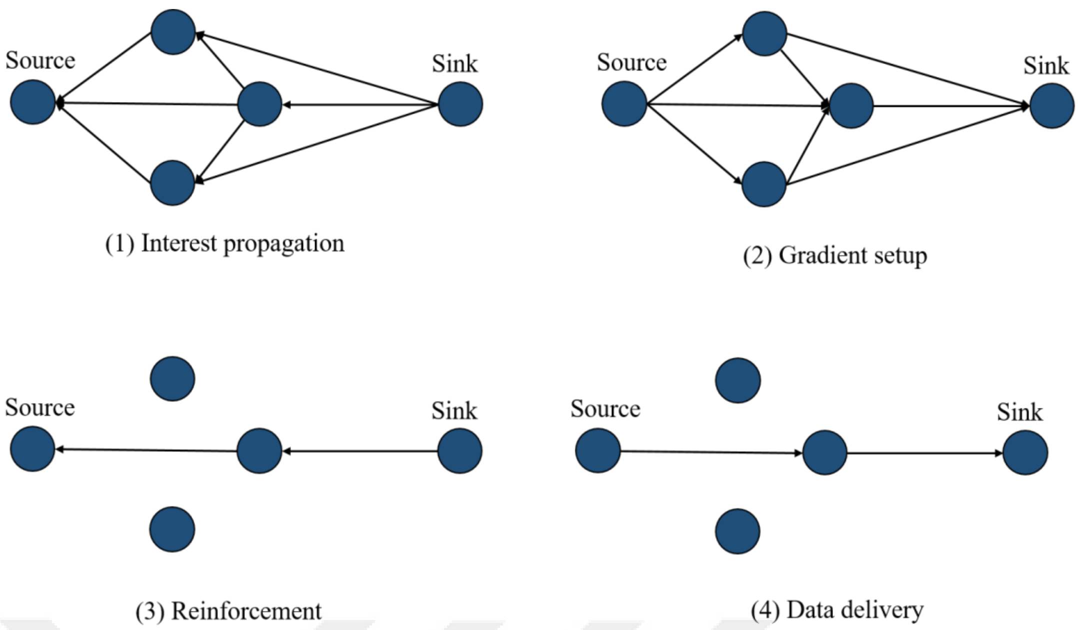


Figure 1. 9 An example of Directed Diffusion

Then, the sink/BS can choose a particular path and reinforce it to transmission data. There are some rules to select this path like the lowest delay, link quality. After the specific node selected to transmission the interest message, the transmission is occurred between the node and the source to reinforce the path. All sensor members through this path transfer the data to reinforce this path and a path creates between the source and the sink/BS. In addition, the transmission path can be changed dynamically. Since, the sink broadcast interest message periodically, the new path, and the gradient will be established. Furthermore, there is a negative reinforcement to remove the current path. While the sink/BS may select the different path or multiple paths along the reinforcement stage.

The DD algorithm also is scalable. During the network lifetime, a node may die or a new node added, the DD algorithm tries to reinforce and create new gradients. As mentioned before, the DD algorithm provides a data delivery guarantee. Also, it has multiple path between the sink/BS and sensor nodes, it provides fault tolerance when a node dead or a node loses the path (Kiani, 2014).

1.3.2. Hierarchical-based networks

As the flat-based network is suitable for applications that detect events the hierarchical-based networks are most suitable for monitoring applications. Flat-architecture suffers from overload and uneven energy consumption. The close node to the sink/BS consumes more energy and die faster. As a result, it decreases the scalability and connectivity of the network. Hierarchical or cluster-based routing protocols, originally proposed in wireline networks, are well-known techniques with special advantages related to scalability and efficient communication (Sohraby et. al., 2007). As such, the concept of hierarchical routing is also utilized to perform energy-efficient routing in WSNs. In a hierarchical architecture, higher energy nodes can be used to process and transfer the information while low energy nodes can be used to perform the sensing in the proximity of the target. It achieved by the clustering approach. There are two types of sensor nodes in this type of network, cluster head (CH) and cluster member (CM). CHs communicate with each other and transfer the data packet to the sink/BS. Also, CMs have communication with own CH. So it's an efficient way to reduce energy consumption. Cluster heads decrease the number of transmission to the sink/BS.

Generally, hierarchical-based networks have two-layer for routing, the first layer is used the cluster head selecting and the second layer provides routing for the network (Jamal et. al., 2004). Also, these routing protocols suitable for large-scale networks. In hierarchical-based networks, prolong network lifetime by some data aggregation and reduction. Most of the well-known hierarchical based routing protocols include LEACH (Heinzelman et. al., 2000), PEGASIS (Lindsey et. al., 2002), TEEN (Manjeshwar et. al., 2001a), and APTEEN (Manjeshwar et. al., 2002b) are described in the below section. Many approaches proposed recently that we will describe them in the related works section.

1.3.2.1.Low Energy Adaptive Clustering Hierarchy (LEACH)

LEACH is a clustering algorithm that developed by (Heinzelman et. al., 2000). LEACH algorithm is an example of the hierarchical algorithms. The main goal of this algorithm is to increase network lifetime, control the overhead and data aggregation. The algorithm is based on clustering and cluster head (Figure 1.10). In this algorithm, all of the nodes are chosen in a homogeneous manner, which means that they have the same battery capacity and communication range. The operation of the LEACH is conducted in two stages, that is the setup phase and the steady stage. In the setup phase, cluster and cluster head selecting. In the steady stage, the data transfer to the base station is implemented. Cluster head and cluster members are selected in the setup phase. In the steady-state, data transfer to the base station is implemented.

The cluster head is selected by using Equation 1.1. For each node, a random value between 0 and 1 is assigned. If the random value is less than the node threshold value T (n) that node is selected as the cluster header.

$$T(n) = \begin{cases} \frac{p}{1-p*(r \bmod \frac{1}{p})} & if\ n \in G \\ 0 & otherwise \end{cases} \tag{1.1}$$

Where**:**

- n is the given number of nodes
- p is the prior probability of a node being elected as a cluster head
- r is a random number between zero and one
- G is the set of nodes that were not chosen as CHs in the last 1/P rounds

During the steady phase, the selected cluster head broadcasts the advertisement message. In this stage, the non-cluster node gets a cluster head id. In addition, a radio signal strength (RSSI) value is used in order to detect sensor nodes neighbors. Also, the time-division medium access (TDMA) schedule avoids a collision in inter-cluster communication. After forming clusters, TDMA schedule allocated to transmission then cluster heads aggregate received data and send this data to the base station. Also, the CDMA schedule used in the LEACH algorithm inside the cluster to communication between neighbor members.

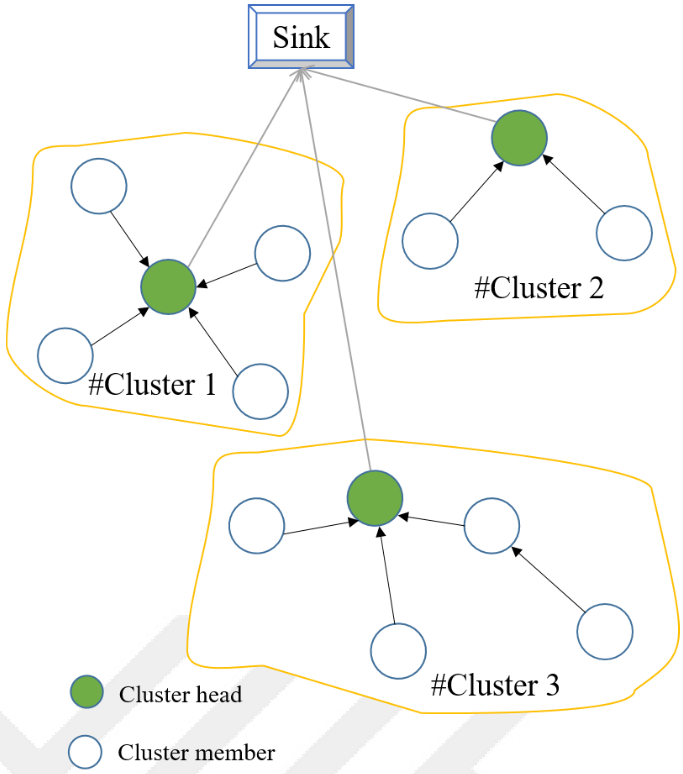


Figure 1. 10 Architecture of the LEACH algorithm

In the LEACH algorithm, there are no criteria to limit the G set. G set represents the non-cluster head sensor nodes in each round. Node residual energy, the distance between nodes and neighbors can optimize network lifetime in cluster head selection. For instance, the distance between the sensor node and a base station can be more than a threshold value or residual energy of a node can be less than a threshold value. Therefore, it is not efficient to select this node as a CH i.e. cluster head.

1.3.2.2.Power-Efficient Gathering in Sensor Information Systems (PEGASIS)

PEGASIS (Lindsey & Raghavendra, 2002) is one of the clustering algorithms and is an improved version of the LEACH algorithm. The main difference of the ELACH and PEGASIS is in the structure, while the LEACH relies on the cluster-based hierarchical mechanism, the PEGASIS uses a chain structure. The data packets forward among the chain in a static and homogenous network. The main goals of PEGASIS energy efficiency and data aggregation and extend network lifetime. Reduce the packet delay on the way of transmission between sink/BS and sensor nodes is another goal of PEGASIS. The chain performed in a greedy approach because nodes select the close neighbor as the next destination (Akyildiz et. al., 2010). The algorithm assumed that each node has global knowledge about neighbors and also each node keeps the next and previous neighbors' information. As a result, nodes just communication with their closest neighbors.

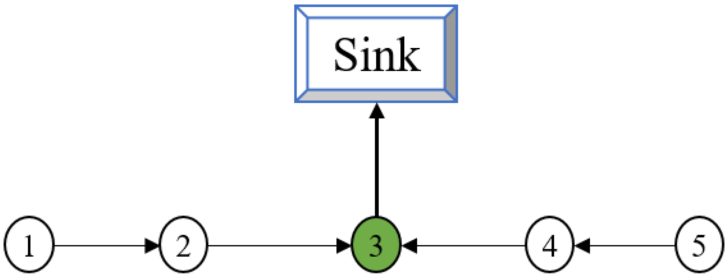


Figure 1. 11 Data transmission in the PEGASIS

In the chain mechanism, there is a leader that is responsible for transfer aggregated data to sink/BS (Figure 1.11). The network has one leader node, while the leader node in each round may change in a random position. This is incurred to avoid a fast drain battery level (Lindsey et. al., 2002). According to Figure 1.11, node 3 selected as a leader node. Nodes 1 and 2 from the left side send data to the leader and nodes 4 and 5 communicate with the leader from the right side of the chain. While node 1 sends data packet to node 2, that node aggregate node 1 and own packet then sends aggregated data to the leader (node 3) also from the right side this happens. After this, the leader node transmit aggregated data to the sink/BS. In this mechanism, data fusion helps to reduce energy consumption also when a node dies, the chain will be reconstructed. Radio Signal Strength Indicator (RSSI) measures the distance to all its neighbors. Overhead in the PEGASIS algorithm is lower than the LEACH algorithm. The PEGASIS uses CDMA in chain communication.

1.3.2.3.Threshold Sensitive Energy-Efficient Sensor Network Protocol (TEEN)

TEEN is illustrated by (Manjeshwar et. al., 2001a). The TEEN algorithm is a hybrid of hierarchical-based and data-centric protocols. It is event-based delivery in the network (Manjeshwar et. al., 2001a). This algorithm is suitable for applications that monitor sudden changes in the network such as temperature, humidity, magnetic flux. TEEN applies multiple levels of hierarchy in the network. The data packets initially collect aggregate and transfer to the first layer of hierarchy, then this data transfers to a higher level of cluster head. The cluster head in each cluster based on periodic time changed.

As mentioned before the TEEN algorithm is suitable for sudden changes, TEEN provides two thresholds for sensed attributes, hard threshold (Ht) and soft threshold (St). The TEEN algorithm is not counties active; the status is changed by the threshold value. The hard threshold keeps the minimum value of an attribute. Also, the soft threshold is a value to check the little or no change in the sensed attribute. The threshold values reduce the number of transmissions in the network. For instance, the sensor nodes check two threshold values in transmission. If the hard threshold value is exceeded, the sensor node checks the soft threshold and if the difference between the two values is greater than the soft threshold, then it should send data to CH. Consequently, hard and soft threshold limits the transmissions between the sensor node and CH.

1.3.2.4. Adaptive Threshold Sensitive Energy Efficient Sensor Network Protocol (APTEEN)

APTEEN is introduced (Manjeshwar et. al., 2001b). APTEEN algorithm is suitable for periodic reports applications. APTEEN uses a TDMA schedule to transmission. The main difference between TEEN and APTEEN algorithm is just the periodic reports. Therefore, the APTEEN algorithm also uses hard and soft thresholds furthermore uses the time schedule frequently. APTEEN is able to support three types of queries: historical queries, one-time queries, persistent queries. Queries information is in below:

- Historical queries: this is used for analyzing the past data values
- One-time queries: take snapshot from network to collect the whole information
- Persistent queries: monitor events for a period of time

The simulation results illustrated in (Manjeshwar et. al., 2001b) between TEEN, APTEEN, LEACH. The results performed that TEEN is much better than LEACH. TEEN and APTEEN have low overhead in comparison with LEACH.

1.3.2.5. Hybrid Energy-Efficient Distributed Clustering (HEED)

HEED is one of the clustering algorithms that are described in (Younis & Fahmy, 2004). In this algorithm, all nodes are again homogeneous. In HEED, different cluster heads will be selected based on the residual energy and the density of nodes. In HEED, links between the nodes are symmetric. Non-uniform energy consumption can occur for all nodes due to the nodes' responsibilities, processing, and communication capability. Prolonging network lifetime, controlling the overhead and making compact well-distributed clusters are the goals of the HEED algorithm.

In this algorithm, cluster head selection is a hybrid of residual energy and communication cost. Average minimum reach-ability power (AMRP) is the mean of the minimum power levels. This value is required by all M nodes within the cluster range to reach u.

$$AMRP = \frac{\sum_{i=1}^{M} MinPwr_i}{M} \tag{1.2}$$

Here, Equation 1.2 and 1.3 are originally referenced in Younis and Fahmy's work (Younis et. al., 2004). For Equation 1.2, $MinPwr_i$ is the minimum power level required by a node v_i to communicate with a cluster head u such that$1 \leq i \leq M$. For every step, a probability is set in such a way that it describes the initial percentage of cluster heads among all n nodes asC_{prob}. The residual energy of the node as $E_{residual}$ and maximum energy of the supplied battery as E_{max} are arguments of the cluster probability that are calculated for every node as follows:

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{max}} \tag{1.3}$$

Cluster head sends a message to its neighbors with the following format: [Nodeid, selection-status, Cost]. The selection status consists of two values, $Tentative_{CH}$ or$Final_{CH}$. Here, $Tentative_{CH}$ means that at each epoch the node needs to be the cluster head. If CH_{prob} has reached the value 1, this will be changed to $Final_{CH}$ mode, and until CH_{prob} becomes less than 1 the node selection status will be kept as$Tentative_{CH}$. $Tentative_{CH}$ node can become a regular node at a later iteration if it finds a lower cost cluster head.

A node is considered as "covered", if it heard a message as $Tentative_{CH}$ or $Final_{CH}$. If it did not hear any messages, then the node is considered as "uncovered", and it announces a message with the selection status $Final_{CH}$. By considering, the explanations mentioned above, low cost and high residual energy are important factors for the selection of the cluster head in the HEED algorithm. Yet, the HEED algorithm has some limitations. For instance, at each round, selecting the cluster head causes more overhead in the network. Consequently, energy dissipation increases and decreases during the lifetime of the network.

Also, in the HEED algorithm, a node can be considered as uncovered status and its status may change from $Tentative_{CH}$ to $Final_{CH}$. In this case, more clusters with only one cluster member will be created.

1.3.3. Geographical-based networks

The coordination of the sensor node can be suitable information in routing protocols. In a location-based routing algorithm there are two basic assumptions, first is each node has information about its own position and neighbors, also sensor nodes receive messages about the position. The nodes in this kind of network equipped by location estimate devices such as a global positioning system (GPS). The sensor nodes forward the data packets on the base of the location and destination. As a result, the data packet can be broadcast just for a specific region and this is lead to consume energy efficiently.

This kind of routing algorithm has low overhead in comparison to other types of routing protocols. The source nodes can be requested on-demand to save energy. Also, the location-based routing protocols provide sleep and awake scheme. Mobile sensor nodes with GPS can be effective in routing algorithms because location information shared between sensor nodes and neighbors easily find each other to forwarding data packets. As in geographical based routing algorithms, location information shared between each other, sensor nodes estimate the distance between two sensor nodes and can predict energy consumption. Some famous routing protocols in this type such as; MECN (Rodoplu et. al., 1999), SMECN (Li et. al., 2001), GEAR (Yu et. al., 2001), GAF (Xu et. al., 2001), SPAN (Chen et. al., 2002) described in the next subsection.

1.3.3.1.Minimum Energy Communication Network (MECN)

MECN is a routing algorithm that is a type of location-based protocol (Rodoplu et. al., 1999). The sensor nodes in these protocols are mobile and equipped with low-power GPS and motion detection sensors. MECN algorithm provides peer-to-peer communications. Due to this algorithm are a position-based network and the sensor node is capable of self-reconfiguration, this algorithm provides the minimum energy solution. This algorithm is suitable for the network designed for minimum energy. Also, MECN increases the whole network's lifetime. MECN algorithm considers a subnetwork in the protocol that requires less power for transmission. This leads to finding a path with a minimum cost between the sink/BS and sensor nodes. MECN has two phases:

Phase 1: At the first MECN, apply a local search on the network to find potential neighbors. Then the algorithm made a two-dimensional enclosure graph. The enclosure graph includes all closest neighbors to transfer, edges and connections between nodes.

Phase 2: the unnecessary link on the graph is removed and the result is a minimum energy topology. The cost of each node established a route to sink/BS.

According to Figure 1.12, the MECN algorithm provides a relay region. Some nodes in the relay region surround an area. Relay region provides a low cast transmission in comparison to direct transmission. Relay region created when MECN apply the enclosure graph. MECN has not a good performance in static networks, due the neighbors of sensor nodes are static and this is a reason for draining the energy supply.

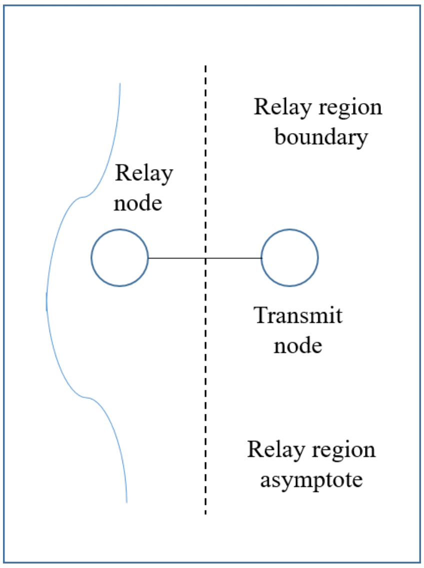


Figure 1. 12 Relay region of transmit-relay node pair (i, r) in MECN

1.3.3.2.Small Minimum Energy Communication Network (SMECN)

SMECN is an improved version of MECN that provide adaption with node failure (Li et. al., 2001). It is considered the obstacles between two pairs of nodes. SMECN divides the network into some sub-networks while the subnetwork in MECN in few more than SMECN. The smaller sub-networks have a limited number of edges. The number of hops in forwarding data packets is less than the MECN algorithm. Simulation results prove that SMCEN consumes less energy than MECN. The maintenance costs are also lower than the MECN (Kiani, 2014).

1.3.3.3.Geographic and Energy Aware Routing (GEAR)

GEAR is introduced by (Yu et. al., 2001). The main idea of GEAR is to reduce broadcast by choosing a specific region to broadcast, unlike Directed Diffusion (DD). Each node has information about its location, remaining energy level also these values about the neighbors. Also, the assumption for this type of network is mobile sensor nodes.

Therefore, the node can estimate transfer cost with two-parameter: residual energy and distance to destination. Also, the GEAR algorithm learned a cost to destination. The learned cost is the value of the estimated cost around holes in the network. The hole in the network is created when if a sensor cannot find a destination to transmit the data packet. The estimated cost and learned cost will be the same when there is not any hole in the network.

The GEAR algorithm consisted of two-phase; the first phase is to transmit the data packet in the target region. The second one is to forward the data packet to the destination. In the first phase, there is a greedy scheme to select a neighbor. When a sensor node receives a data packet, checks if there is a closer node to the target region than itself, selects that node as next hop. Else, there is a hole around the network. The second phase, use recursive geographic forwarding or restricted flooding to transmit within the region. The recursive geographic forwarding is good when the network is deployed densely otherwise, the flooding method is suitable. In the recursive geographic scheme, the target region divided into four regions, and the sensor node transmits the data packet to sub-regions.

1.3.3.4.Geographic Adaptive Fidelity (GAF)

GAF is introduced by (Xu et. al., 2001). It is an energy-aware location-based algorithm. The main goal of GAF is to turn off a node if it is equivalent from a routing perspective also, it adjusts routing fidelity by node deployment density. Fidelity is persistent connectivity between communicating nodes. In the GAF there are virtual grids in the result of the nodes' geographic location information. The grid should be in the size of r. (Equation 1.4). Where R is the communication range of nodes.

$$r \leq \frac{R}{\sqrt{5}} \tag{1.4}$$

The geographic location information helps nodes to associate itself in the grid. In the square virtual grid, only the sensor node from adjacent grids can connect. The goal of the GAF is that keeps only one sensor node in every grid but maintain connectivity in the network in a distributed way. The nodes in the same grid have the same cost of packet routing. While each node in a grid can communicate with all nodes in the adjacent grid. For instance, if there are four nodes in the same region, three of them can sleep. In this way, GAF prolongs the network lifetime. Sensor nodes in GAF have three states:

- Discovery state: discover the neighbors in the grid
- Active state: participate in forwarding data packets
- Sleeping state: turn off transfer mode

When a node in the discovery state, the node broadcast discovery message with neighbors. The discovery message contains, ID, grid ID and estimated node active time (enat) and node current state. When a node in discovery state it has a time as T_d . Since each node has estimated active time (enat), the rank of the node can be determined. In this state the node wait to receive discovery messages, if receive a message from a node with a high ranking number in this the node switched to sleep mode. Because the node in the active state has a higher ranking from the discovery state. Also if the node does not receive discovery message then it switches to active state on the expiration time of T_d . In the active state, the node participates in the backbone node to transmit data packets and maintain connectivity. Each node has a T_a and broadcast discovery message with intervalT_d. When active time expired, that node switches to discovery state. As GAF divided the network in virtual grids, except just a single node of each grid others turn off transfer mode to reduce energy consumption. These nodes set a time T_s as sleep time, after time expiration changes state as discovery.

As a result, the GAF performance is based on the accurate location of each node. In this way, the sensor nodes in this protocol should have a GPS. In cases, the backbone nodes have not to error in location, the decisions of GAF in high accuracy. Consequently, the GAF successfully extends network lifetime.

1.3.3.5. An Energy-Efficient Coordination Algorithm (SPAN)

This algorithm is illustrated by (Chen et. al., 2002). The main goal of the SPAN is the decrease in energy consumption. SPAN also use backbone nodes to forward data packets. So, the neighbor's information is necessary. SPAN is focused on network connectivity. There is a problem in backbone capacity due to the path between sensor nodes and sink/BS is decreased in communication. This is one of the congestion problems in the backbone. SPAN gives this problem because in the communication it considers the backbone capacity and connectivity.

Span broadcast messages to change or create a new path in communication. Also in this, algorithm sensor nodes have information about their own location and neighbors. Actually, SPAN keeps the position of the two or three hops neighbors. Each node broadcasts a HELLO message. The hello message keeps this information:

- Node status:
- The list of backbone node it is connected to

- Neighbors list

The broadcasting HELLO message leads to each sensor node has information about the neighbors and know backbone nodes. In addition, the HEELO message gives a broad knowledge of neighbors to nodes. In addition, if in the communication of backbone neighbors two or more than two nodes cannot transfer information, those nodes eligible to be in the backbone.

1.4.Energy efficiency on Wireless Sensor Networks

As mentioned in the characteristic of the WSN's section, the sensor nodes have equipped by the limited battery. It is important to design efficient routing algorithms. In most of the application of the WSN, the network life is a vital basic. Actually, if there is enough sensor node to communicate the network is alive, the network has efficiency, otherwise, with most of the dead node; the network does not have efficiency.

Battery supply recharge or change is almost hard. First the large size of the network and second the environment. Most of the wireless sensor networks consist of thousands and thousands of sensor nodes. Also, as an advantage of the WSNs is easy to install in harsh environments, like in mountain, under the asphalt, military areas. As a result recharge and change battery is impossible.

There are schemes to reduce energy consumption in the WSN. These techniques should trade-offs with other sensor nodes elements. When an algorithm cares about energy consumption, it should consider other network factors such as security, packet delivery, and throughput. Most of the researches in energy efficiency try to balance all factors of the wireless sensor networks. In this research (Anastasi et. al., 2009) generally divided energy-efficient schemes into three types:

- Duty cycling
- Data-driven approaches
- Mobility

Duty cycling methods focused on subsystems. This scheme also has the operation on the transceiver in the communication. This algorithm strives to employ most of the energy supply by applying sleep and wake modes in periods. Generally, the sensor should become ready to sense as soon as possible. The algorithm decides when a node switches the mode from awake to sleep and vice-versa. The duty cycle has two different approaches. The first one is possible to exploit node redundancy. In this approach, the algorithm selects only a subset of the network and keep nodes in active mode to provide connectivity. Remain nodes changes to mode to sleep to save energy.

In the data-driven approach of energy efficiency schemes, there are two ways to effect on the WSN for energy efficiency. In a first way, the algorithm sort unnecessary nodes and deactivate them in transmission to the sink/BS. In a second way, maintain the accuracy of the subsystem sensor node at a reasonable level reduces energy consumption. The first way dissolved unnecessary samples, while the second way decreases the energy consume in sensing.

1.5.Metaheuristic algorithms

A large and growing body of literature has investigated; many authors have proposed new metaheuristic algorithms. These algorithms are fast becoming a key instrument in complex optimization problems, so, the search space is an important issue in each complex problem. By the increasing dimension of a problem, search space increases exponentially so the complexity of a problem will be increased (Winston et. al., 1990; Yao et. al., 1997). In some problem functions, the dimension is constant. Metaheuristic algorithms can provide reasonable solutions with an acceptable time.

Metaheuristic algorithms may not always guarantee the best solutions and sometimes the findings solutions maybe not acceptable. Therefore, the performance of the developed algorithms may vary from problem to problem. An algorithm that is very successful in the solution of a problem may not be good for another problem solution at the same time. The metaheuristic algorithm is executed based on random inputs and received outputs and independently from the problem (Mirjalili et. al., 2014).

The classification of metaheuristic algorithms can be applied based on various criteria such as single or multi (population) based solutions. In a population-based algorithm whole of the population have an effect on the output. Whereas in a single based algorithm a single solution along search iteration is evolved (Talbi, 2019). As explained above, randomness, reasonable solution, acceptable response time and different solutions for a problem in each run are the characteristics of metaheuristic algorithms (Mirjalili, 2015). In the literature, population-based studies have been conducted more (Talbi, 2019). They are used in many areas of science and engineering such as engineering design, machine learning, system modeling, industry, planning in routing problems (Talbi, 2019). The main purposes of metaheuristics are solving problems faster, solving large problems and obtaining robust algorithms (Talbi, 2019).

As is known, metaheuristic methods are from the family of optimization algorithms. These algorithms are categorized into two groups of exact and approximate algorithms. Exact algorithms are capable of finding the optimal solution in a precise manner, but they are not efficient enough for difficult or hard optimization problems and their execution time expands exponentially with the dimensions of the problems.

Approximate algorithms are capable of finding good (near-optimal) solutions in a short time for difficult/hard optimization problems. Heuristic and metaheuristic algorithms are in the category of approximate algorithms. The heuristic algorithm is problem-dependent, whereas the metaheuristic algorithm is a problem-independent technique. Besides, heuristic algorithms may be trapped in local optima. However, metaheuristic algorithms avoid trapping in local optima by exploration and exploitation concepts (Mirjalili et. al., 2014). These concepts are very important in each metaheuristic algorithms because should have a tradeoff between two of them. Exploration means to find the best solution in the area(s) and exploitation refers to focus on the best solution area(s) to reach the best solution. Taken as a whole, in the initial iteration power of exploration must be increased following it the power of exploitation gradually will increase.

Metaheuristic algorithms are generally classified into three types: evolution-based, physics-based, and swarm intelligence methods. The evolution-based algorithms (EA) originated by nature. Evolutionary algorithms for solving a given problem in a search space initially starts with a random population (set of solutions). In these methods, the best solution in each process has an effect on the next generation of individuals. The most popular algorithm in this category is the Genetic Algorithm (GA) (Holland, 1992) that has been inspired by Charles Darwin's theory of evolution.

GA mimics generation reproduction and includes selection, crossover, mutation and the elitism of generation phases.

Another algorithm based on EA is Differential Evolution (DE) same as GA mimics evolutionary theory but there are some differences between them such as the selection operators (Storn & Price, 1997). Evolutionary Programming (EP) emphasized the development of behavioral models like phenotype, hereditary, variation (Yao et.al., 1999; Fogel, 1998). Biogeography-Based Optimizer (BBO) is inspired by natural laws. It explains migration between islands and factors that why and which island is the best election for migration (Simon, 2008).

The physics-based method generally mimics physical rules and biological processes of nature. In this kind of algorithms, physical rules have more effect in search spaces. There are more well-known physics-based algorithms. One of the most popular algorithms is Gravitational Search Algorithm (GSA) (Rashedi et. al., 2009) that has been inspired by gravity law and mass interactions. Black Hole (BH) algorithm mimics the black hole phenomenon (Hatamlou, 2013). Chemical Reaction Optimization (CRO) (Lam et. al., 2009) simulated the molecules interact with each other through a sequence of elementary reactions. Central Force Optimization (CFO) is based on the metaphor of gravitational kinematics (Formato, 2007).

The swarm-based methods in other meaning Swarm Intelligence (SI) algorithms based on group behaviors. These types of algorithms consist of a group of simple particles and homogeneous members that interact with each other and their environment. SI based algorithms impact on agents that cooperative in local search space and collective behavior of all agents caused to reach convergence near to the best solution. In these algorithms, each agent is expected to cooperate with other agents. Particle swarm optimization (PSO) is the most popular algorithm in this category that was presented by (Eberhart & Kennedy, 1995). PSO was inspired by the social behavior of birds, in PSO there are communication channels between particles. They move on search space. The fitness function will determine the best solution. PSO can provide good solutions to various optimization problems.

In the literature, different algorithms were proposed in the various fields using this method. One of them is the best route-finding problems. The pathfinding in the routing based problems is a critical issue. Especially, when the paths are frequently changed, finding the fix and best routes is difficult. Also, other parameters such as power, delay, delivery rate and etc. are effective in the decision mechanism. In these category problems, PSO can build optimal solutions and routing paths.

Another study is Ant Colony Optimization (ACO) that simulates behaviors of ants in finding the path in foraging (Dorigo & Di Caro, 1999). ACO impacts some factors that each ant need to find path according to the other ants' experience in the path. ACO is an excellent example of SI algorithms. ACO has been used in many fields such as routing in wireless sensor networks.

In this category, another study is an Artificial Bee Colony (ABC) that mimics the behavior of the honeybee. In this colony, foraging is one of the main behaviors (Karaboga & Basturk, 2008). In ABC there are two types of bees; employed and unemployed bees. These bees are responsible for the search for rich food sources. Bees benefit from the experiences of other bees in finding a resource. For example, if a specific location has good resources, they will try to go there. In reverse situations, if resources are being low there, the location will be abandoned. Firefly Algorithm (FA) inspired by the behaviors of fireflies (Yang, 2010). In this algorithm, the flashing characteristics of fireflies attract other fireflies. The flashing method has a message, usually used for sending a signal to the opposite-sex of Firefly in the colony. Cuckoo Search (CS) algorithm mimics the cuckoo's behaviors in the nest (Yang et. al., 2009).

They have one instinctive behavior that distinguished cuckoo from other birds. Cuckoos lay their eggs in the nest of other birds (of other species), this is known as obligate brood parasitism (Mohamad et. al., 2013). If the host bird discovers that an egg is not its own, it tries to destroy the egg or migrate to another nest. Usually, cuckoo tries to have some changes on own egg to be similar to the host's nest egg.

Grey Wolf Optimizer (GWO) is another algorithm that is based on the SI category and has been introduced in (Mirjalili et. al., 2014). GWO mimics the social habits of grey wolves and focuses on the grey wolves hunting mechanism. It is based on the leadership structure of grey wolves. In GWO there is four types of wolves; alpha, beta, delta, and omega. All other wolves were given the name Omega. Authors in GWO supposed alpha, beta and delta wolves have a better knowledge of hunt position. Omega wolves update their own position based on three wolves defined smartest to encirclement hunt. In literature are many studies that apply GWO to find an optimal solution in various problems.

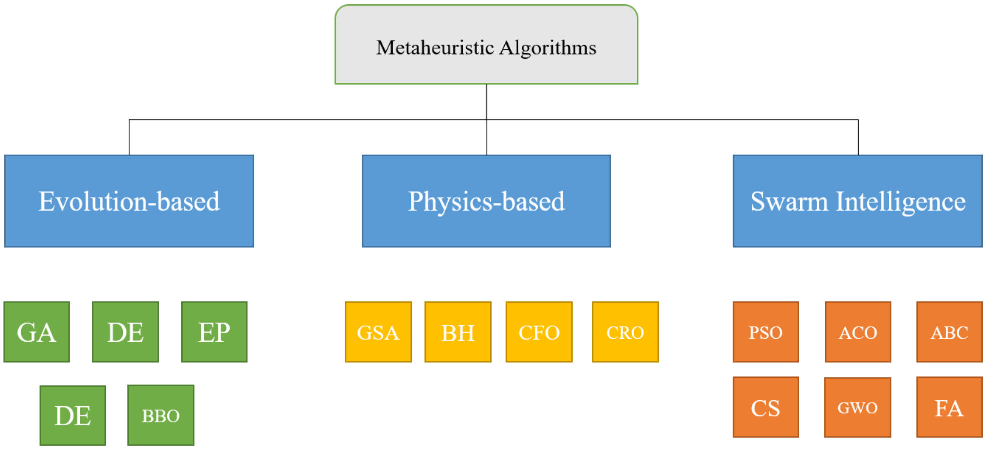


Figure 1. 13 Metaheuristic algorithms categories

As mentioned above, most studies in the field of the metaheuristic algorithms have focused on exploration and exploitation. The algorithms proposed in this field maintain the balance between these two phases. For example, in GWO, the authors claimed that there is a tradeoff between them. As mentioned before, there are four types in wolves in GWO. Omega wolves update own position based on the first three wolves position. The purpose of these updates is to catch the prey. These three wolves are located at the first three levels of the hierarchy and the remaining wolves (omega) are defined at the fourth level. In this mechanism, since the position updates of the wolves at the second layer (fourth and next levels-omega) depend only on three wolves, they can be densely settled in the same or in certain regions during the prey catching process. In this case, the mechanism of escape prevention may not work well.

This thesis is proposed two new metaheuristic algorithms inspired by the GWO method. One of them is an algorithm that is Expanded Grey Wolf Optimizer (Ex-GWO), the position of the wolves in the first layer (alpha, beta, and delta) and the wolf /wolves previously selected and updated in the second layer are used to update each current wolf own position. The other recommended method in this thesis is an Incremental-based Grey Wolf Optimizer (I-GWO) algorithm. In the working mechanism, the position update of each wolf is related to wolves that were selected and updated before. In other words, the n-1 wolves' positions are considered in the position update of the nth wolf. Both, there is a high probability of finding good solutions for a variety of complex problems. Also, it can find global solutions quickly in few iterations. The convergence rate to the global solution in the Ex-GWO method is lower than in the I-GWO method but it has more balanced behavior and performance in many problem-solving. These two algorithms are described in more detail in section 2.2 of the thesis.

1.6.Dissertation Structure

This thesis proposed two routing protocols and three clustering methods. The routing protocols achieved the energy-efficient with metaheuristic algorithms. Also, the writer of this thesis proposed two improved metaheuristic algorithms. Generally, the first improved algorithm inspired by grey wolf optimization (GWO) (Mirjalili et. al.,2014). The proposed algorithms named incremental grey wolf optimization (I-GWO) and expanded grey wolf optimization (Ex-GWO). In addition, the author modified the ant colony optimizer (ACO) (Dorigo & Di Caro, 1999) and adapt the ACO with the routing protocols rules.

This thesis consists of five chapters. The first chapter explained the wireless sensor network characteristic as the main subject of this thesis on energy efficiency, give a brief literature review. The remainder of this thesis is organized as below (Figure 1.14):

- Chapter 2: Discusses the related works on energy-efficient routing protocols and the prolonging of the network lifetime schemes.
- Chapter 3: This chapter presents the proposed method. In chapter 3 also there are 5 subsections. Subsection 3.1 describes the HEEL algorithm. 3.2 introduces the EEHRSN algorithm. Also in 3.3, there is the I-HEEL. The improved version of the HEEL algorithm with metaheuristic algorithms (GWO, I-GWO and Ex-GWO). Section 3.4 proposes new efficient routing protocols inspired by metaheuristic algorithms (I-GWO and Ex-GWO). Finally, section 3.5 describes the efficient routing protocol with the ACO algorithm.
- Chapter 4: Concludes the thesis and suggests future work on the wireless sensor networks.

Finally, there are references for the thesis.

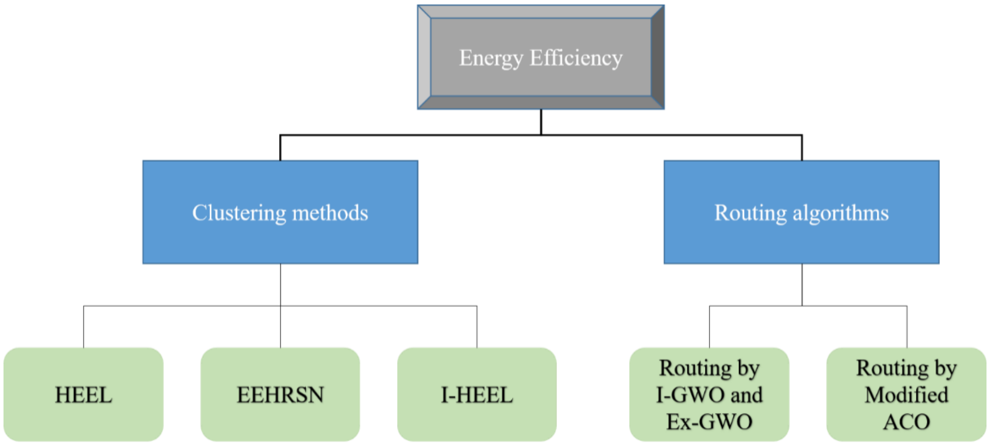


Figure 1. 14 Thesis taxonomy

CHAPTER TWO: RELATED WORKS

The main issue in WSN is energy consumption. WSN consists of many sensor nodes and usually, they are deployed unmethodical. The principal rule of wireless sensor networks transmitting the collected data to sink/BS. When a node has a sensed data it needs a protocol to transmit it to neighbor nodes or sink/BS. The network needs a routing protocol to find a path. Most routing protocols transmit a data packet in single or multiple hops. As WSN has limited battery supply, the routing protocol should provide an energy-efficient path.

The characteristic of a sensor node in the network may be different such as mobile or static sensor nodes. Mobile sensor nodes are embedded with a module to find own position, that it consumes the energy. The sensor nodes in the network are heterogeneous or homogeneous nodes. In heterogeneous networks the energy level, transmission range is different. In addition, the wireless sensor network may be large or small scale. The application of wireless sensor network varies from each other. As a result, the routing algorithm should cope with some challenges. The efficient energy routing algorithm is one way to improve network lifetime.

The clustering method achieves energy efficiency using the hierarchical network structure. There are several studies that are conducted in this field of study. LEACH is one of the famous protocols that is described by (Heinzelman et. al., 2000). In addition, most of the new protocols are based on LEACH such as NR-LEACH (Al-Baz & El-Sayed, 2018), MELEACH (Chen & Shen, 2008), HHCA (Lee & Kao, 2016), TL-LEACH (Loscri et.al. 2008), ModLEACH (Mahmood et. al., 2013), and LEACH-B (Tong & Tang, 2010), ERPLBC-CS (Omar & Khedr, 2018), EACLE (Yanagihara et. al., 2007). We focus on the hierarchical-based network. Section 1.3.2 explained the main famous algorithms such as LEACH (Heinzelman et. al., 2000), PEGASIS (Lindsey & Raghavendra, 2002), TEEN and APTEEN (Manjeshwar et. al., 2001a; Manjeshwar et. al., 2001b), and HEED (Younis & Fahmy, 2004).

The main concept of cluster head selection protocols is to decrease energy consumption. In the clustering method, the choice of the cluster head is important. The main rule of the hierarchical network is based on cluster heads. The aim of the CH is to collect data from the CMs and send the collected data to a sink/BS.

In order to send data, a protocol is necessary. The base station collects the data from cluster heads, so it is necessary to do extra precision in data delivery to reduce the lost packet rate. In most studies, the data has sent to sink/BS hop by hop. Therefore, the base station must have some information concerned with the received packets.

In addition, heterogeneous sensor networks are discussed. Some of the well-known algorithms suchlike SEP (Smaragdakis et. al., 2004), DEEC (Qing et. al., 2006), EECH (Kumar et. al., 2009a) are explained in the following. In a heterogeneous sensor network, there are different sensor node types; the difference is on the battery supply, transmission range. As well as this type of network has categorized in two-level and three-level clustering. These protocols attempt to enhance the network lifetime, refine stability and minimize latency.

Metaheuristic algorithms are finding the best solution from a search space. These algorithms are used in most of the engineering sciences. In this thesis, these algorithms to find an optimal cluster head. In addition, finds the path between destination and source at minimum cost. The authors proposed two metaheuristic algorithms based on the GWO algorithm. A modified ACO algorithm is also proposed in optimal pathfinding. The second section explained the ACO, GWO, MGWO (Rajakumar et. al., 2017), EGWO (Joshi & Arora, 2017). The I-GWO and Ex-GWO algorithms are also explained in section 2.2.5.

In the third section of this chapter, the routing and clustering method with metaheuristic algorithms explained. There are popular routing protocols with metaheuristic algorithms such as EBAB (Wang et. al., 2009), LTAWSN (Mohajerani & Gharavian, 2016), ACOHCM (Jiang & Zheng, 2018), sensor node localization by GWO (Joshi & Arora, 2017), GWO-LPWSN (Rajakumar et. al., 2017).

2.1. Energy-Efficient Routing Algorithms

2.1.1. Modified LEACH (MODLEACH)

MODLEACH stands for a modified LEACH algorithm that is introduced by (Mahmood et. al., 2013). This algorithm is a variant version of the LEACH algorithm. The MODLEACH algorithm's goal is to improve throughput and network lifetime.

Two methods have proposed in the MODLEACH to promote the performance of the LEACH algorithm. The first method is an effective solution to replace the cluster head. The second method is dual transmitting power levels. In the first method, authors have proposed a threshold value in CH formation in the setup phase. If the existing CH residual energy is more than the threshold value and no more energy is being spent, then it will remain as the same CH for the upcoming round.

In addition, the MODLEACH has controlled the amplification energy. The equal energy consumption for both sensor nodes located in near and far from, the base station is considered. Data transmission is divided into three modes as below:

- Intra cluster transmission
- Intercluster transmission
- CH to sink/BS transmission

Therefore, the intracluster transmission is proposed for cluster members and cluster head communication inside a cluster. In addition to that, the inter-cluster transmission is proposed for the transmission between two cluster heads. High energy amplification used for cluster heads to communicate with other cluster heads. Low energy amplification is used for cluster member's communication and cluster head. So, the required energy for communication is different.

2.1.2. Node Ranked LEACH (NR-LEACH)

Authors in this algorithm improved the LEACH algorithm to increase network lifetime (Al-Baz & El-Sayed, 2018). Node ranked LEACH (NR-LEACH) algorithm is a new clustering method for hierarchical network structure. NR-LEACH is based on two parameters to choose cluster head: path cost and the number of links between nodes. The authors claimed that the NR-LEACH algorithm has good performance in a large-scale network.

The first parameter in NR-LEACH is path cost, which calculates received signal strength and residual energy for each node. NR-LEACH also focuses on the node’s residual energy similar to other leach algorithms. NR-LEACH select cluster head from each node's ranking. Nodes rank will be obtained according to its position, residual energy, and connection links. The rank function selects a node with the highest rank. Equation (2.1) is node rank function:

$$NR(n_i) = PO(n_i) \times \alpha \times \sum_0^j NR(n_j) \frac{1/d_{out}^{ij}}{\sum_{k \in NH} d_{out}^{jk}} + (1 - \alpha) \tag{2.1}$$

In Equation (2.1), NH is the set of neighbors for the k, d_{out}^{ij} is the distance from node j to node i, $PO(n_i)$ is the current energy of the node i and α is damping factor. Authors in (Al-Baz & El-Sayed, 2018) considered damping value equal to 0.85. The initial rank of each node considered is one. NR-LEACH algorithm repeats this function to each node until it reaches convergence score. This algorithm avoids random selecting cluster heads.

2.1.3. an Efficient Load-Balanced Clustered Routing Protocol (ERPLBC-CS)

This algorithm proposes an efficient load-balanced clustered routing protocol (Omar & Khedr, 2018). Authors in this algorithm combine a novel clustering method with a compressive sensing (CS) theory. CS theory is one of signal processing techniques for transferring sufficient information with low power consumption. Cluster head election in this algorithm depends on residual energy and concentration degree of sensor nodes.

The concentration degree of any node is the number of sensor nodes that can sense during the same round. The base station (BS) broadcast E^r is the average residual energy of network, if residual energy of each node is bigger than E^r that specific node's calculated weight from residual energy and concentration degree. After this, that node sends that weight with its own ID to BS. BS decides to which sensor node will be selected as cluster head by choosing the maximum weight. Then, a sensor node that is selected as CH will broadcast to its neighboring nodes to inform them. In this method, the authors claimed that the proposed algorithm can produce and balance energy consumption in addition to the prolongs lifetime of the network.

2.1.4. A two-levels hierarchy for low-energy adaptive clustering hierarchy (TL-LEACH)

There are several other algorithms which are the modified versions of the LEACH algorithm (Loscri et. al., 2008). TL-LEACH is another algorithm that is proposed with a two-step hierarchy.

- A top cluster head as primary cluster head CH_i
- A second cluster head as secondary cluster head CH_{ij}
- and Simple node as SN

In this method, cluster heads do not send the data directly to the base station. There exist two data aggregation methods. At first, SNs send data to CH_{ij} s, and then CH_i collect the aggregated data from CH_{ij}. In TL-LEACH, two-step cluster head selection causes extra overhead.

2.1.5. Energy-Aware Clustering Scheme with Transmission Power Control for Sensor Networks (EACLE)

The authors in this algorithm proposed a new energy-aware clustering method (Yanagihara et. al., 2007). In this method, control transmission power to improve the network lifetime. EACLE tries to build a tree scheme by sink node and sensor nodes. In EACLE, there are two power transmission modes "power low" and "power high". Power low is used for packet transmission inside of the cluster between CM and CH. Power high is used between CHs transmission. Authors in EACLE noted that there is not any other information about the distance between nodes in data packets. The connection between cluster heads is a mesh network connecting. In the routing step, only cluster heads have routing information, so the connecting tree makes fast between CHs.

2.1.6. Stable Election Protocol (SEP)

This algorithm is a method to select the cluster head in heterogeneous wireless sensor networks (Smaragdakis et. al., 2004). The authors in this method proposed a stable election protocol (SEP). The aim of the SEP algorithm to increase stability. The cluster heads collect and aggregate data from appropriate cluster members to transmit to the sink/BS. The sensor nodes have a different energy level; some of the sensor nodes have more energy than others. Normal and advanced nodes are two-level heterogeneous sensor nodes. All the sensors deployed randomly and are not mobile. The SEP algorithm assumed cluster head selection is random. Only the remaining energy of each node has an effect on the selection cluster head. The advantage of the SEP algorithm is that each node does not need global knowledge of energy. Just SEP algorithm performance is low in multi-level heterogeneity (Tanwar et. al., 2015).

The energy of advanced nodes is more than the normal nodes in (α) percentage. All sensor nodes in this network have static probability to elect as CH. SEP algorithm is based on a hierarchy structure and is inspired by the LEACH algorithm (Heinzelman et. al., 2000). SEP can prolong the stability period and use a probability distribution for a node to become cluster-head stands the residual energy of the node. In cluster head selection SEP threshold follow Equation 2.2, 2.3 as in below, m the fraction of the total numbers of n, these are the advanced node, and the rest of $n \times (1 - m)$ is normal nodes.

$$T(S_{nm}) = \begin{cases} \frac{p_{nrm}}{1 - p_{nrm} \times (r \bmod \frac{1}{p_{nrm}})} & S_{nrm} \in G' \\ 0 & Otherwise \end{cases} \tag{2.2}$$

$$T(S_{adv}) = \begin{cases} \frac{p_{adv}}{1 - p_{adv} \times (r \bmod \frac{1}{p_{adv}})} & S_{adv} \in G'' \\ 0 & Otherwise \end{cases} \tag{2.3}$$

Where:

- r indicates the current round
- G' refers to nodes that have not been selected as CH in the last $\frac{1}{p_{nrm}}$ round
- $T(S_{nm})$ refers to n*(1-m) normal nodes
- G'' refers to a set of the advanced node that has not been chosen as CH in the last $\frac{1}{p_{adv}}$ round
- $T(S_{adv})$ refers to advanced nodes

Then, if the threshold value bigger than a random number between [0, 1] the node selected as CH.

2.1.7. Distributed energy efficient clustering (DEEC)

The DEEC algorithm is inspired by the LEACH algorithm (Qing et. al., 2006). One of the main differences between DEEC and LEACH is in the type of sensor nodes, unlike the LEACH algorithm, and DEEC is focused on heterogeneous sensor nodes. This algorithm suitable for multilevel heterogeneous sensor networks, also it has cluster head mechanism for transmission data between sensor nodes and sink/BS. The clustering method in this algorithm depends on the ratio between the remaining energy of each node and the mean value energy of the network. The mean value for in r_{th} time interval calculated from Equation 2.4:

$$\bar{E}_i(r) = \frac{1}{N}\sum_{i=1}^{N} E_i\,(r) \qquad (2.4)$$

Equation 2.5 gives the optimal no of cluster head:

$$p_{opt}\sum_{i=1}^{N}\frac{E_i(r)}{\bar{E}_i(r)} = N_{opt} \qquad (2.5)$$

If a node has higher energy than others, that node is selected as a cluster head. Here weighted election probability (WEP) for the multi-level heterogeneous network is in below:

$$p_{(si)} = \frac{p_{opt}(1+a_i)}{(N+\sum_{i=1}^{N} a_i)} \qquad (2.6)$$

Where:

- $\bar{E}_i(r)$ indicates the mean value of energy at r_{th} round
- p_{opt} denotes the optimal no of cluster head
- a is the times more energy of advanced nodes than the normal nodes

The network is deployed with *N* sensor nodes in $M \times M$ square are. There is *N* number of normal sensor nodes and *m* number of advanced sensor nodes. The normal sensor nodes' initial energy is E_0. The advanced sensor nodes' energy is α times more than others are. So, the advanced nodes energy is $E0(1+a)$ and normal nodes $N(1-m)$ initial energy. Equation 2.7 calculates the total energy of the network.

$$E_{totoal} = N(1-m)E_0 + NmE_0(1+\alpha) = NE_0(1+\alpha m) \qquad (2.7)$$

Equation 2.7 is related to a two-level heterogeneous network, equation 2.8 is the total energy of multi-level heterogeneous networks.

$$E_{total} = \sum_{i=1}^{N} E_0(1+\alpha_i) = E_0(N + \sum_{i=1}^{N} \alpha_i) \tag{2.8}$$

The DEEC has a threshold value to select the optimal node as CH. In the two-level heterogeneous network, follow Equation 2.9.

$$p_i = \begin{cases} \frac{p_{opt}E_i(r)}{(1+am)\bar{E}(r)} & if\ s_i\ is\ the\ normal\ node \\ \frac{p_{opt}(1+a)E_i(r)}{(1+am)\bar{E}(r)} & if\ s_i\ is\ the\ advanced\ node \end{cases} \tag{2.9}$$

In this algorithm, advanced node dies rapidly faster than other nodes as residual energy decreases. DEEC only considers the sensor's energy as a source of heterogeneity. As a result, the DEEC algorithm selects a cluster head with weighted election probability and the remaining energy of each node that chose the higher energy. Also, data transmission is done in a single hop. In addition, it is suitable for applications with large areas and monitoring environments.

2.1.8. Energy Efficient Heterogeneous Clustered Scheme (EECH)

This method is proposed a new energy-efficient clustering scheme for heterogeneous sensor nodes (Kumar et. al., 2009a). EECH based on clustering methods with two phases; setup and steady phase. The first step is calculating the optimal cluster numbers. The side length of sensing and the total number of sensor nodes are two parameters in the cluster selection. These sensor nodes have a difference in three factors in a hierarchical topology. The difference in computational, link and energy. So, three different weighted probability is defined. The network includes normal, advanced and super nodes. The main goal of EECH is to improve network lifetime, reliability, and packet delivery, also decrease the latency.

At the first step of EECH, the optimal number of cluster heads computed as follows, first based on distance give a value for k_{opt} (Equation 2.10) then p_{opt} (Equation 2.11) computed:

$$k_{opt} = \sqrt{\frac{n}{2\pi}}\sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}\frac{M}{d_{BS}^2} \tag{2.10}$$

Then:

$$p_{opt} = \frac{K_{opt}}{n} \tag{2.11}$$

Where:

- n is the number of nodes that distributed uniformly
- ε_{fs} is multipath fading energy consumption
- ε_{mp} is free space energy consumption and d is the distance between the nodes
- M refers to the area of the network

In the next step, the weighted probability for three types of sensor nodes respectively calculates as follow:

$$p_n = \frac{p_{opt}}{1+m* (\alpha+m_0*\beta)} \tag{2.12}$$

$$p_a = \frac{p_{opt}}{1+m* (\alpha+m_0*\beta)} * (1+\alpha) \tag{2.13}$$

$$p_n = \frac{p_{opt}}{1+m* (\alpha+m_0*\beta)} * (1+\beta) \tag{2.14}$$

p_{opt} replaced in this equation, then for normal nodes cluster head selection follows the below equation. Similarly, we find the threshold cluster head for advanced and super nodes:

$$T(sn) = \begin{cases} \frac{p_n}{1-p_n*\left(r*mode\frac{1}{p_n}\right)} & \text{if } s \in G' \\ 0 & \text{otherwise} \end{cases} \tag{2.15}$$

After these phases, the CH is selected and members of the cluster get correspond cluster head id to start transmission. In this algorithm, CHs transmit the data packets to sink/BS in a single hop. The energy efficiency of the EECH algorithm is very good in comparison of LEACH and SEP algorithms. Also, (Kumar et. al., 2011b) proposed a new improved version of the EECH. The cluster head selection as the same as EECH. There is also a new multi hop routing algorithm (MCR) to improve stability. Furthermore, this method control load balancing.

2.2. Metaheuristic Algorithms

2.2.1. Ant Colony Optimization

This metaheuristic algorithm is a nature-inspired method for solving complex and global optimization problems (Dorigo & Di Caro, 1999). ACO identified artificial ants inspired by the real-life behavior of ants. Although ants are not insects living in the logic of the herd, they are assisting and guiding each other using chemical deposits called pheromones and as a result, behave as if they are a smart colony. In other words, they live in a collective wisdom model.

The working mechanism of this algorithm is an iteration-based method like most of the metaheuristic based algorithms. So, some ants, which are defined in first of simulation, are trying to find possible solutions for a relative problem. Indeed, each ant presents one solution that in the last of each iteration; the ACO selects the best solution of the possible founded solutions. In each iteration, each ant leaves pheromone in every path of choice, so the pheromone values of the selected paths will be constantly up to date.

Furthermore, at the end of each iteration, the pheromone values collected at all edges are evaporated to prevent local optimal problems. Each ant starts from nest toward food and backward to the nest. Each round of go and back is called tour. After completing one tour of all the ants, one iteration is considered complete. In the ACO the problem modeled as a search for the best path by constructing a path-graph. (Kulkarni et. al., 2010). Usually, each ant prefers the path with more pheromones. Thus, fewer pheromones will be left on long paths. Indeed, the source node periodically sends a forward ant and the ant selects the next-hop node according to the pheromone intensity. In fact, the ACO uses the rule of pseudo-random proportional.

ACO algorithm, as mentioned before, can be applied to different complex problems in various search space areas to find one or more solutions based on the essence of the metaheuristic methods. One of the application areas of the ACO algorithm can be routing problems. ACO routing algorithms perform in the well-distributed model, also support adaptively, robustness, and scalability. Therefore, it has been frequently used in wireless sensor networks in recent years. In most studies in the literature, methods for efficient use of energy and network are generally presented. In some studies, this algorithm has been used by targeting other parameters efficiently.

Just as ants have laws in nature to find food and find a way to get food, the ACO algorithm is an artificial example of simulating ants' behavior. As can be seen in Figure 2.1, the ants find a way to move from the food to the nest. This path may initially have errors such as failure to get nest such as arriving delay, obstacles. An ant first begins its movement toward food individually, leaving in its path some of the chemical named pheromone on the ground. In other words, this chemical is a kind of markup. If the ant is looking for food afterward, reach for food by following these chemicals or markups.

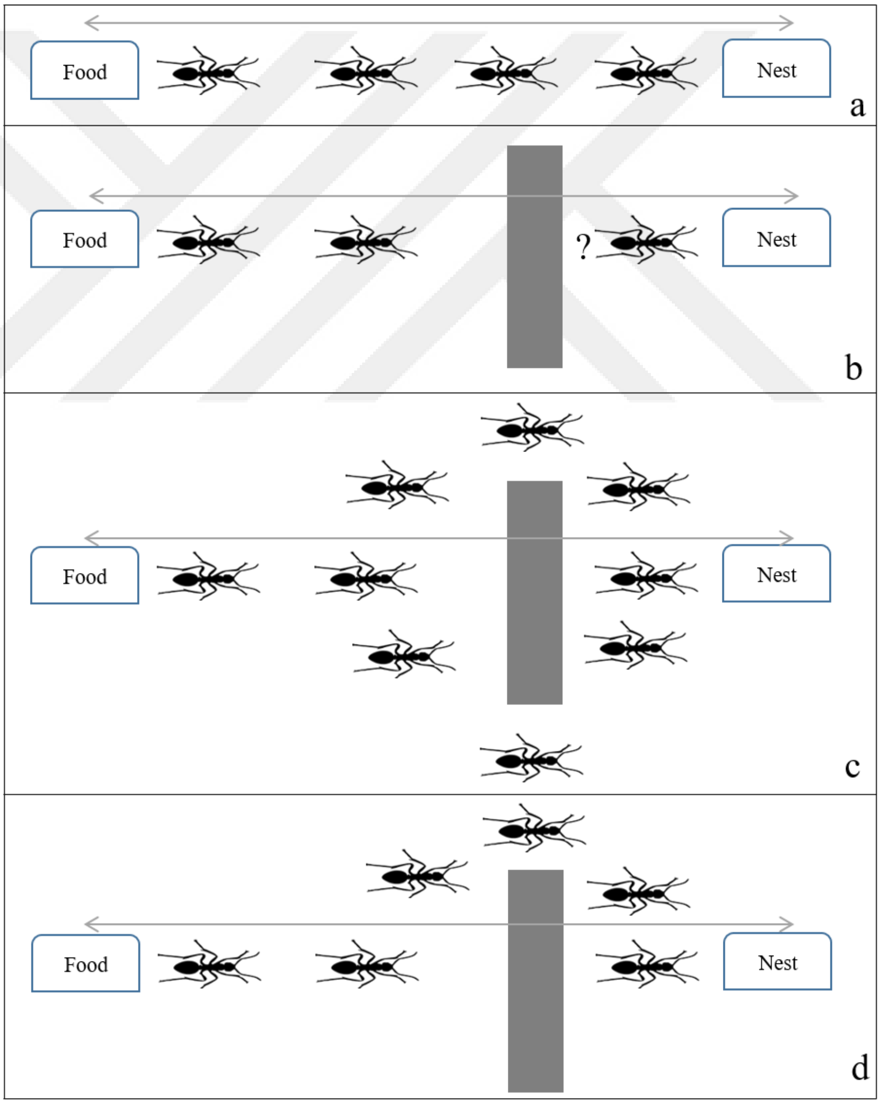


Figure 2. 1 Ant's behavior

However, after the path to reach the food, the ant must have a return path to the nest, which will also follow the nest with the pheromone. Then ants discovered a safe way from the nest to the food. Other ants will also look for food, they will naturally continue this path, along the way an ant may choose another path, which depends on the amount of pheromone, maybe a new path and may have no pheromones. So a new path from the nest to the food was discovered. However, the path that is closest to the food and chosen by the largest number of ants will be selected as the best path.

The ACO inspired by real ants foraging behavior, as they find and selects the best and optimal path between nest and food. As one of the concepts of metaheuristic algorithms to work on search space, the ants explore the environment to find the best path. The mathematical model of ACO at the first applied in "Travelling salesman problem" (Stützle & Dorigo, 1999). In the following, the mathematical formulas explained. Mathematical models such as update pheromone value in each path, how to select next hop and evaporation mechanism.

Consider, the ant k tries to find a path between source i and destination j. An equation named as Equation 2.16 gives function probability.

$$p_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in N_i} [\tau_{il}]^{\alpha} \cdot [\eta_{il}]^{\beta}} & if\ k \notin N_i \\ 0 & otherwise \end{cases} \tag{2.16}$$

Where:

- $p_{ij}^{k}(t)$ refers the probability at the t_{th} iteration of the algorithm that ant k located in destination *i* and choose source *j* to move
- τ_{ij} indicates the pheromone value
- η_{ij} is the heuristic value
- a and β are the control parameters of pheromone and heuristic value respectively
- N_i is the set of sources that ant k visited

This function is used to select the next destination. For all the possible destinations, that ant *k* has to choose destination *j*. The higher value obtained of this value is selected as the next destination of the ant *k*. The pheromone value for the ant *k* in edge *i* and *j* calculated from Equation 2.17:

$$\tau_{ij}(t) = \rho\tau_{ij}(t-1) + \Delta\tau_{ij} \tag{2.17}$$

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \tau_{\ ij}^{\ k} \tag{2.18}$$

Where:

- ρ indicates a control parameters to influence on evaporation

After each tour, an amount of pheromone is added to the path visited by ant k. the heuristic value is related to the application. Maybe it is the Euclidean distance between two cities, the energy value of the destination. In the “traveling salesman problem” (Stützle et. al., 1999) the heuristic value is the distance between two cities by Equation 2.19 where d_{ij} is the distance between the city *i* and *j*.

$$\eta_{ij} = {}^{1}/_{d_{ij}} \tag{2.19}$$

The above function is also called the heuristic function. In this example, the “Traveling salesman problem” in each round is the reverse ratio of the distance to the edge of each edge. Generally, for each fixed amount of pheromone, whatever the shorter the edge length has a higher density.

The ACO algorithm is applied in many fields of the science and engineering. Examples of application in graph coloring (Costa & Hertz, 1997). Nondeterministic tree-search procedures (Maniezzo, 1999), solve the quadric assignment problem (Maniezzo & Colorni, 1999), vehicle routing problem in (Gambardella et. al., 1999), energy efficient routing protocol in WSN based on ACO (Camilo et. al., 2006), energy efficient coverage in WSN with three pheromone (Lee et. al., 2011), efficient data aggregation in the WSN (Lin et. el., 2012).

2.2.2. Standard Grey Wolf Optimization

In 2014, (Mirjalili et. al., 2014) published a paper in which they described a new metaheuristic optimization algorithm as Grey Wolf Optimization (GWO). GWO has been simulated based on the social behavior and leadership hierarchy of grey wolves. Grey wolf group living and hunting habitat is unique to these kinds of animals. Grey wolves’ groups usually consist of 5-12 wolves. Each group consists of four types of grey wolves namely alpha (α), beta (β), delta (δ), and omega (ω) as shown in Figure

2.2. For example, 2 members in a 5-members group are Omega wolf. It must be said that each of the omega wolves may have different tasks in the pack.

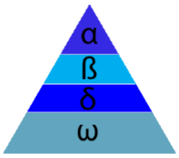


Figure 2. 2 Grey wolf hierarchy

In groups, each of these type of grey wolves has a different responsibility. Alpha (α) wolf has a powerful effect in the group - the leader. A decision about a hunt, sleep location, wake up time are responsibilities of alpha wolf. This decision is dictated to the group. Only the Alpha Wolf can mate. It does not matter whether this wolf is stronger than others are. Group management is more important than pack power in hunting. Beta (β) wolf is in the second layer of pack hierarchy.

These type of wolves known as co-leaders in the group. Beta wolves help alphas in decision-making and can be good substitutes for alphas. Beta wolves dictate instructions to wolves in the lower hierarchy. Delta (δ) considered in third level of hierarchy. A Delta wolf must be follow the instructions from upper level wolves; alpha and beta. Delta is the last wolf that can eat hunt. It is important to say that if the delta wolf does not exist in a group, the group encounters internal chaos and problems. In the lowest layering of hierarchy is omega (ω) grey wolves. If the wolf does not include any of the above types, wolf is omega. In addition, one of the interesting habits of grey wolves is group hunting. The authors of the GWO algorithm suggested a mathematical model inspired by grey wolf life, so, the GWO algorithm mimics some real behaviors such as the encircling, hunting and attacking the prey.

2.2.2.1.Mathematical model in encircling the prey

In hunting, wolves encircle the prey. Mathematically, it is modeled as given in Equation 2.20 and 2.21. Thanks to this, the hunt with the new location of the gray wolves will be surrounded.

$$\vec{D} = \left|\vec{C} \cdot \overrightarrow{X_p}(t) - \vec{X}(t)\right| \tag{2.20}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \tag{2.21}$$

Where, t is the current iteration, $\vec{A}$, $\vec{C}$ are coefficient vectors, $\vec{X}$ is the position vector of the grey wolf and $\overrightarrow{X_p}$ is the position vector of the prey. $\vec{A}$, $\vec{C}$ and $\vec{a}$are calculated as Equations 2.22, 2.23, and 2.24 respectively.

$$\vec{A} = 2\vec{a} \cdot \overrightarrow{r_1} - \vec{a} \tag{2.22}$$

$$\vec{C} = 2 \cdot \overrightarrow{r_2} \tag{2.23}$$

$$\vec{a} = 2\left(1 - \frac{t}{T}\right) \tag{2.24}$$

Where, $\vec{a}$ is linearly decreased from 2 to 0 over the courses of iteration. It is used to get closer to the solution range. $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$ are the random vectors in range of [0, 1].

2.2.2.2.Mathematical model for hunting mechanism

Grey wolves have an ability to surround prey position. In the mathematical model is supposed that there is no idea about the position of the prey. So alpha, beta, and delta have better knowledge about the prey's position. Indeed, alpha (the first best solution), beta and delta are the three best candidate solutions. Omega wolves renew their positions according to the wolves in the upper layer. The related Equation 2.25, 2.26, and 2.27 are proposed in this regard.

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1} \cdot \overrightarrow{X_\alpha} - \vec{X}\right|,$$

$$\overrightarrow{D_\beta} = \left|\overrightarrow{C_2} \cdot \overrightarrow{X_\beta} - \vec{X}\right|,$$

$$\overrightarrow{D_\delta} = \left|\overrightarrow{C_3} \cdot \overrightarrow{X_\delta} - \vec{X}\right| \tag{2.25}$$

And

$$\overrightarrow{X_1} = \vec{X}_a - \overrightarrow{A_1} \cdot \overrightarrow{D_\alpha} \ ,$$

$$\overrightarrow{X_2} = \vec{X}_\beta - \overrightarrow{A_2} \cdot \overrightarrow{D_\beta} \ ,$$

$$\overrightarrow{X_3} = \vec{X}_\delta - \overrightarrow{A_3} \cdot \overrightarrow{D_\delta} \tag{2.26}$$

Then

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1}(t) + \overrightarrow{X_2}(t) + \overrightarrow{X_3}(t)}{3} \tag{2.27}$$

2.2.2.3.Mathematical model for attacking prey

It is worth mentioning here exploration and exploitation are important in metaheuristic algorithms. GWO tries to tradeoff between these two phases. In GWO, $\vec{a}$ value in each iteration decreased from 2 to 0. Actually, $\vec{A}$ value is also decreased by$\vec{a}$. The value $\vec{A}$ is important to a grey wolf. When |A|<1 forces the wolves to attack the prey. Otherwise |A|>1 wolves try to search other prey. That proves the exploration and exploitation concepts. $\vec{C}$ Parameter provides random values in each iteration.

2.2.3. Modified GWO Algorithm (mGWO)

In (Mittal et. al., 2016) authors have proposed a new version of GWO. This method focused on proper balance between exploration and exploitation. Modified GWO algorithm enhance the exploration process by decreasing the value of $\vec{a}$ based on the Equation 2.28. That way, by decreasing $\vec{a}$ the value $\vec{A}$ is decreased too. So mGWO can balance in exploration and exploitation to find global minimum with fast convergence speed. mGWO uses exponential decay function of $\vec{a}$ over each iteration. Where, T denotes the maximum number of iteration and t is the current iteration.

$$\vec{a} = 2\left(1 - \frac{t^2}{T^2}\right) \tag{2.28}$$

In this algorithm modification just occurred in $\vec{a}$. This modification of GWO is classified in updating mechanism (Joshi & Arora, 2017). The mGWO was tested using 23 benchmark function and compared with other metaheuristic algorithms. Their results show that the mGWO can find optimal solutions in 10 benchmark functions.

2.2.4. Enhanced Grey Wolf Optimization (EGWO)

The authors in (Joshi & Arora, 2017) introduced a novel hunting mechanism of GWO. Authors tries to balance between exploration and exploitation. Also they focused on improving convergence rate with better results. EGWO algorithm adjusted $\vec{a}$ parameter; a random vector between 0 and 1. So exploration and exploitation are balanced by parameter$\vec{a}$. Authors claim that adjusting in $\vec{a}$ parameter maintain exploration to prevent getting trapped in local optima. In EGWO algorithm, other wolves except alpha (α) update position for hunting just only by leader wolf position. As in the GWO, mentioned alpha (α) has better knowledge about the potential position of prey. EGWO uses alpha (α) position to hunting habit in pack. Hunting mechanism is achieved from Equation 2.29, 2.30, and 2.31.

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1} \cdot \overrightarrow{X_\alpha} - \vec{X}\right| \tag{2.29}$$

$$\overrightarrow{X_1} = \vec{X}_a - \overrightarrow{A_1} \cdot \overrightarrow{D_\alpha} \tag{2.30}$$

$$\vec{X}(t+1) = \overrightarrow{X_1} \tag{2.31}$$

They have tested the proposed algorithm on 25 selected benchmark functions. The authors claimed that the performance of the EGWO is promising in terms of better exploration and exploitation of the search space in comparison with the other some studied algorithms. Their results show that the EGWO can find optimal solutions in 11 benchmark functions.

2.2.5. Proposed Methods Inspired By GWO

In this section, the two proposed methods will be explained in order. The two proposed studies have no superiority over each other and are not recommended for the completion of their deficiencies. In other words, they are not derived versions. These two algorithms can be used to find solutions to different complex problems. Therefore, in a problem where someone maybe not good, the other proposed algorithm might find a good solution.

2.2.5.1. Incremental GWO (I-GWO)

This method is inspired by the classical GWO and EGWO algorithms. In this algorithm is considered that the alpha wolf has best knowledge about the prey position and other wolves in the pack must follow alpha wolf in order. In this method, each wolf updates its position based on all the wolves selected before it. In other words, the n-1 wolves positions are considered in the position update of the n^{th} wolf (Figure 2.3). There is the possibility of finding problem solutions (preys) much faster in fewer iterations. However, they may not always guarantee to find a good solution because they act dependent on each other. Therefore, the speed of growth and the selection of the right places for the first wolf is of great importance.

In this algorithm, the second wolf in the pack updates own position by alpha wolf. Third wolf also updates its position based on alpha and second wolves' positions and so on and so forth. Thus, it is named Incremental Grey Wolf Optimizer (I-GWO) algorithm. In this method, the wolves follow each other according to the defined parameters and method, so they have the chance to find the optimal solution with few iterations. However, this algorithm has a distinct problem. Therefore, the probability of finding best solutions to some complex problems may be low.

This algorithm is greedy and is heavily dependent on the position of the first wolf (alpha). Therefore, a, A and C parameters are important for the alpha wolf to be the best solution. In addition, the best solution of the last iteration of the algorithm may be dependent on the type and dimension of the parameters applied. Therefore, attention is paid to these issues in the proposed method. Otherwise, if the alpha wolf does not have a good position relative to the prey, other wolves update their own positions wrongly and maybe far from the prey. In this case, a good solution can be found if the number of iterations increases. Because the alpha will always be in the best position. Indeed, if the alpha is close to the prey the algorithm can get a solution quickly, however, if the opposite is true, it can still find a good solution, but it needs to increase the number of iterations.

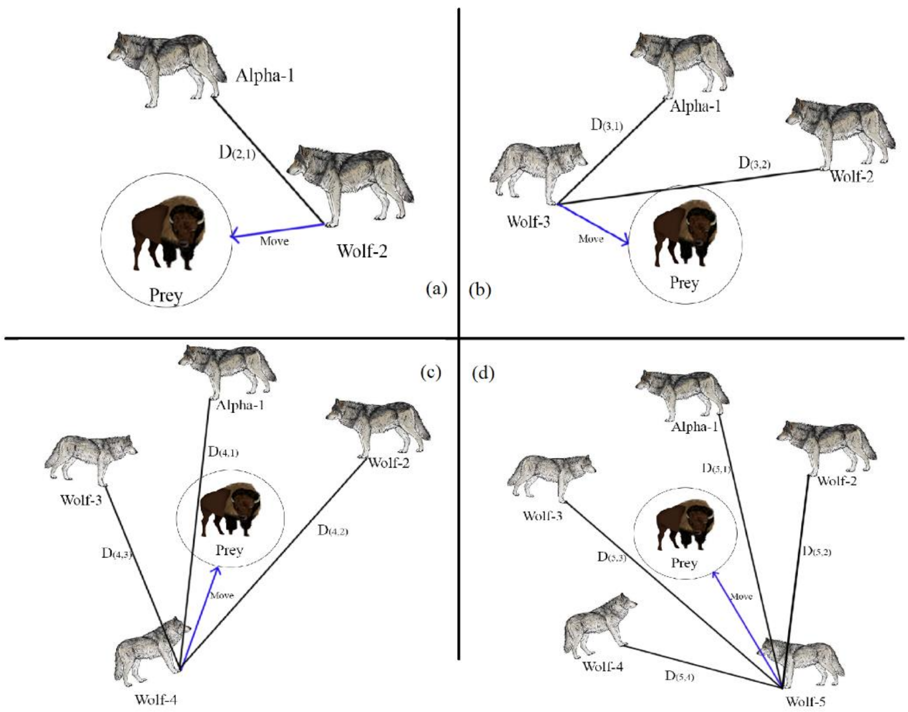


Figure 2. 3 The mechanism of position update for each wolf to catch prey in I-GWO

The proposed algorithm uses some control parameters as are presented in Equations 2.22, 2.23, and 2.28. The effect of $\vec{a}$ is on the range of motion, which directs the algorithm to find the solution (Mittal et. al., 2016). The variable j is defined to increase the number of iterations assigned to the exploration. In hunting mechanism, I-GWO proposed following Equations 2.32, 2.33, and 2.34.

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_\alpha} \cdot \overrightarrow{X_\alpha} - \vec{X}\right| \tag{2.32}$$

And

$$\overrightarrow{X_\alpha} = \vec{X}_\alpha - \overrightarrow{A_\alpha} \cdot \overrightarrow{D_\alpha} \tag{2.33}$$

Then

$$\overrightarrow{X_n}(t+1) = \frac{1}{n-1}\sum_{i=1}^{n-1} X_i(t) \text{ ; n=2, 3,… m} \tag{2.34}$$

Where, *n* is current selected wolf, *m* is the number of wolves in the pack, *t* is iteration and *i* parameter is started from first wolf and continues until the last wolf has been selected and updated before it. I-GWO explained gradually in following flowchart as shown Figure 2.4. The pseudocode of I-GWO is given in Figure 2.5.

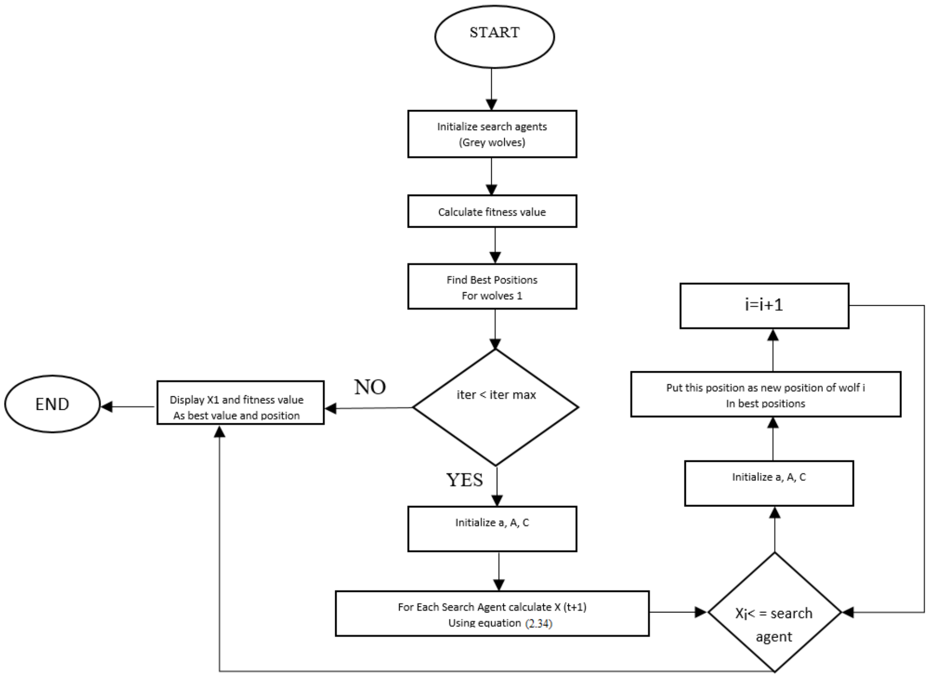


Figure 2. 4 Flowchart of proposed I-GWO algorithm

Initialize the search agent (grey wolf) population X_i (i = 1, 2, . . . , m)

Initialize a, A and C

Calculate the fitness of each search agent

X_a = the best (or dominating) search agent

while (t <Maximum number of iterations)

  for each search agent

    update the position of the current search agent by (2.34)

  end for

  update a by (2.28)

  update A and C by (2.22, 2.23)

  calculate the fitness of all search agents

  update X_i

  insert X_i to best positions table

  $t = t + 1$

end while

Return X_1

Figure 2. 5 Pseudocode of I-GWO algorithm

2.2.5.2.Expanded GWO (Ex-GWO)

As mentioned above, one of the main goals of metaheuristic algorithms is finding the best-optimized solutions. The fitness function is the inseparable base of optimization algorithms. Fitness function value leads the algorithm to reach the best solution maximization or minimization. The population-based optimization algorithm is one type of metaheuristic algorithms. In these algorithms, a random population initially will be generated. This random population (search space) is a set of solutions. In going iteration, this population will be improved. Agents (population) in optimization algorithm try to find best solution from search space.

As discussed in the previous section, this paper is inspired by the GWO algorithm and proposed two algorithms. One of them is expanded version of the GWO and is a population-based optimization algorithm. Generally, In GWO three wolves alpha (α), beta (β), delta (δ) respectively have the highest impact to other wolves that named omega (ω) in the pack. Omega (ω) wolves in the pack must update position according to the alpha (α), beta (β), delta (δ) wolves' position. Omega (ω) wolves have not knowledge about hunt position. They approaching prey based on alpha, beta, delta positions. The GWO considers that they have good knowledge about the prey. Omega wolves have to optimize own position to be close to the prey, so they update own position. It follows that may be omega wolves located in very similar positions or close to each other. In addition, the mechanism to prevent the escape of the hunt may not work well.

The proposed Ex-GWO is a novel hunting mechanism inspired by GWO. In this algorithm, we defined two layers of hierarchy. The first layer consists of three levels that each of alpha, beta and delta wolves are placed in one level. The second layer consists of the other members of the herd (Figure 2.6). The position of the wolves in the first layer (alpha, beta and delta) and the wolf /wolves previously selected and updated in the second layer are used to update each current wolf own position. In this method, the same as GWO, alpha, beta and delta play the role as the main three wolves.

However, the next wolves select and update their positions according to the previous and the first three wolves in each iteration. In the proposed method, to prevent omega wolves from being located in close-up areas like GWO in position updates, is defined a parameter that is called$\vec{a}$, but it may not always be a successful metric. Therefore, a mechanism is recommended that the wolves (Omega wolves) in the second layer follow each other and update own positions. An example of the mechanism is shown in Figure 2.7 for the fifth wolf.

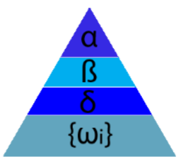


Figure 2. 6 The hierarchy mechanism in Ex-GWO

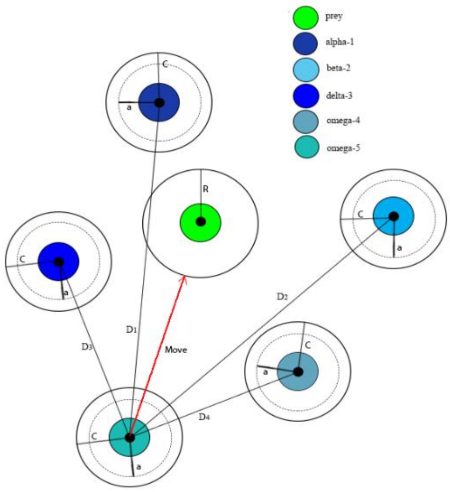


Figure 2. 7 Position updating in the Ex-GWO (e.g., the fifth wolf)

Ex-GWO algorithm uses other positions of wolves in pack, just not uses alpha, beta and delta positions to find the best solution as described in Figure 2.7. So, it is a significant difference between GWO and Ex-GWO in hunting mechanism. The biggest weakness of the GWO algorithm maybe is the combination of Omega wolves. Exploration occurred in half of the iterations and exploitation dedicated in the other half.

The aim of the GWO algorithm is to establish exploration and exploitation phases from the initial iterations. It suggests a balanced performance between the two phases. The time complexity of this algorithm is not good compared to the GWO, but thanks to the balanced behavior mechanism, the probability of finding a good solution is higher. In additional, the wolves in the pack minimize the escape paths of the hunt, hence the hunts can be caught faster. Figure 2.8 is an example of the interaction of wolves with each other, position updates and the hunting sieges.

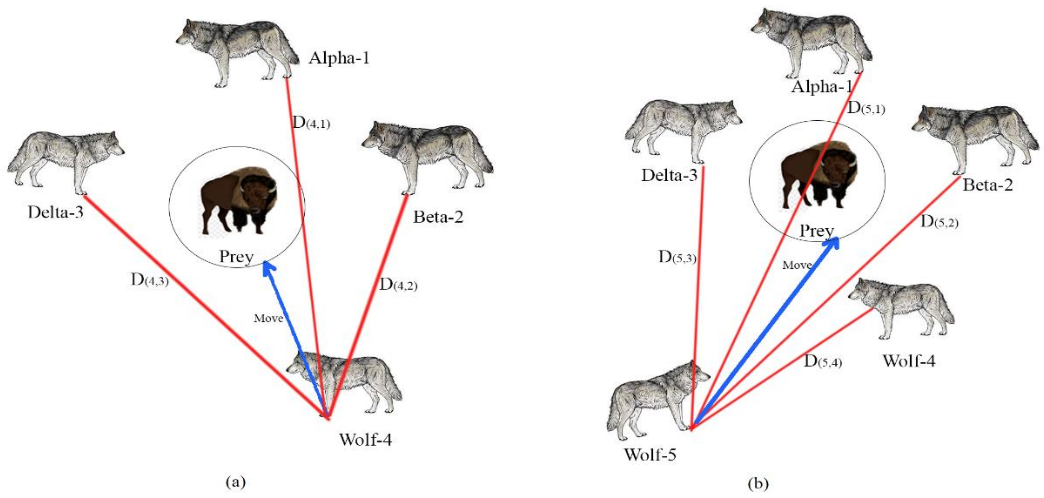


Figure 2. 8 The mechanism of position update for each wolf to catch prey in Ex-GWO. a) Position updating for forth wolf, b) for fifth wolf

In this algorithm, like the GWO, is supposed that first, second and third wolves have better knowledge about prey position. The fourth wolf updates its own position according to the three wolves' position. Evidently, the fifth wolf updates own position based on four wolves (alpha, beta, delta and fourth wolf). The best position of first, second, third and fourth wolves positions help the fifth wolf to update own position. In the same way, the fifth wolf will move to the best position closer to the prey.

This technique is done to each wolf in a pack. The Ex-GWO method represents an innovative alternative to hunting mechanism. In this algorithm, some of the parameters, which were defined in GWO, are revised. Mathematically, a pack has *n* wolves. The first three wolves located in the best position of prey. Remaining wolves update their position in one course of the iteration.

The proposed algorithm is used some control parameters as $\vec{A}, \overrightarrow{C}$ and $\vec{a}$ are presented in Equations 2.22, 2.23, and 2.24. Where, $\vec{A}, \vec{C}$ give direction for the activities of the wolves. Thanks to this, wolves do not always go in the same directions. The effect of $\vec{a}$ is on the range of motion, which directs the algorithm to find the solution. In hunting mechanism, Ex-GWO is proposed based on Equations 2.35, 2.36, and 2.37.

$$\overrightarrow{D_1} = \left|\overrightarrow{C_1} \cdot \overrightarrow{X_1} - \vec{X}\right|,$$

$$\overrightarrow{D_2} = \left|\overrightarrow{C_2} \cdot \overrightarrow{X_2} - \vec{X}\right|,$$

$$\overrightarrow{D_3} = \left|\overrightarrow{C_3} \cdot \overrightarrow{X_3} - \vec{X}\right| \qquad (2.35)$$

And

$$\overrightarrow{X_1} = \vec{X}_1 - \overrightarrow{A_1} \cdot \overrightarrow{D_1}\ ,$$

$$\overrightarrow{X_2} = \vec{X}_2 - \overrightarrow{A_2} \cdot \overrightarrow{D_2}\ ,$$

$$\overrightarrow{X_3} = \vec{X}_3 - \overrightarrow{A_3} \cdot \overrightarrow{D_3} \qquad (2.36)$$

Then

$$\overrightarrow{X_n}(t+1) = \frac{1}{n-1}\sum_{i=1}^{n-1} X_i(t)\ ;\ \text{n=4, 5,… m} \qquad (2.37)$$

Where, *n* is the currently selected wolf, *m* is the number of wolves in the pack, *t* is iteration and *i* parameter is started from first wolf and continues until the last wolf has been selected and updated before it. Finally, wolf *n* updating position from n-1 of previous wolves' position in pack. We believe that in this technique wolves follow a rule in updating own position. Ex-GWO explained step by step in following flowchart as shown Figure 2.9. The pseudocode of Ex-GWO is given in Figure 2.10.

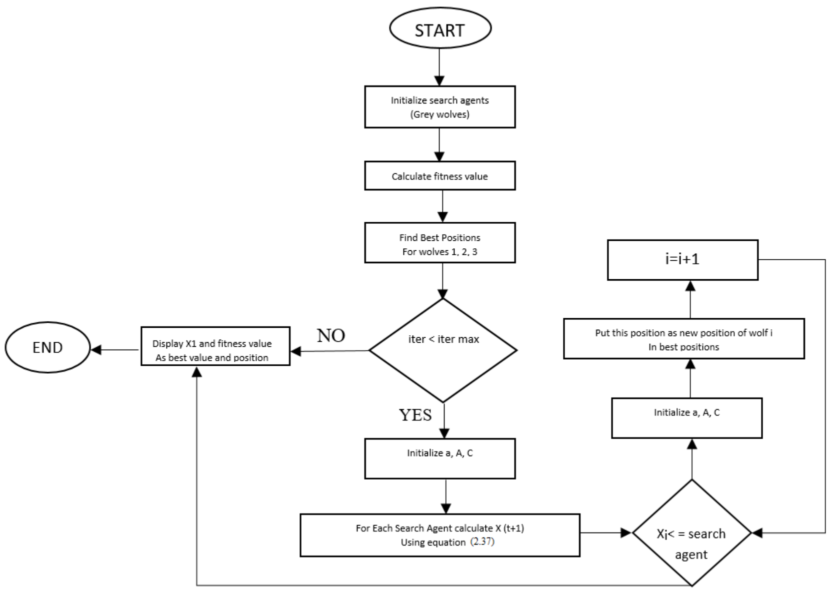


Figure 2. 9 Flowchart of the proposed Ex-GWO algorithm

```
Initialize the search agent (grey wolf) population X_i (i = 1, 2, . . . , m)
Initialize a, A and C
Calculate the fitness of each search agent
X_1 = the best (or dominating) search agent
X_2 = the second best search agent
X_3 = the third best search agent
while (t <Maximum number of iterations)
   for each search agent
         update the position of the current search agent by (2.37)
   end for
   update a by (2.24)
   update A and C by (2.22, 2.23)
   calculate the fitness of all search agents
   update X_1, X_2 and X_3
   insert X_i to best positions table
    t = t + 1
end while
Return X_1
```

Figure 2. 10 Pseudocode of Ex-GWO algorithm

2.3. Energy Efficient Algorithm by Metaheuristic Algorithms

In this section, we review some of the energy-efficient routing algorithms that improved by metaheuristic algorithms. Metaheuristic algorithm inspired by nature. Some of the popular metaheuristic algorithms introduced in the introduction section of this thesis (Section 1.5). These algorithms try to find the best solution to Np-hard problems. In WSN the routing protocols and localization of sensor nodes have been studied with metaheuristic algorithms.

2.3.1. Ant Colony Optimization Router Chip

Since the routing path is important in WSN, (Okdem & Karaboga, 2009) proposed a new method in routing algorithms based on the ACO algorithm. The main purpose of this algorithm is to attain reliable data communication, in addition, to increase network lifetime. They tested their algorithm in the ACO router chip. The mechanism in this algorithm in broadcasting is similar ACK/REQ scheme. When a data packet reaches to destination node sends a data packet as an acknowledgment message. After this, the sensor nodes with high energy levels are chosen for the next hop.

Each ant k tries to find a path from source r to destination s, the probability function is given by Equation 2.38.

$$p_k(r,s) = \begin{cases} \frac{[\tau(r,s)]^{\alpha} \cdot [\eta(r,s)]^{\beta}}{\sum_{r \in R_s}[\tau(r,s)]^{\alpha} \cdot [\eta(r,s)]^{\beta}} & if\ k \notin tabu^r \\ 0 & otherwise \end{cases} \quad (2.38)$$

Where:

- $[\tau(r,s)]$ is pheromone value is calculated
- $\eta(r,s)$ is the heuristic value about energy
- R_s refers to receiver nodes
- $tabu^r$ indicates the list of nodes that node r received data packets previously
- α and β are the control parameters of pheromone trail and heuristic value respectively

Then:

$$\tau(r,s) = \frac{(I-e_r)^{-1}}{\sum_{r \in R_s}(I-e_n)^{-1}} \quad (2.39)$$

Where:

- I is the initial energy
- e_r refers to current energy of receiver node
- e_n indicates to neighbor energy

After whole of ants in the network completed their tour, each ant deposits amount of pheromone. Equation 2.40 gives that it.

$$\Delta\tau^k(t) = {}^{1}/_{J_w^k(t)} \tag{2.40}$$

Where:

- $J_w^k(t)$ refers to the total number of nodes visited by ant k of tour w at iteration t

Also, in the ACO, when a tour completed, should evaporate the deposited pheromone value from arcs. It is done in equation 2.41.

$$\tau_{ij} \leftarrow (1-p)\tau_{ij}(t) \tag{2.41}$$

2.3.2. Life Time Aware Routing Algorithm for Wireless Sensor Networks (LTAWSN)

LTAWSN investigated a new pheromone update mechanism (Mohajerani & Gharavian, 2016). The proposed algorithm achieves the tradeoff between route hops and energy consumption. Because of the importance of energy in wireless sensor networks, the authors consider two energy metrics. In selection next node in routing, a higher probability of sensor nodes that nearer to destination. Also, they used four control parameters in probabilistic decision function. The probability function to choice the next-hop is in bellow:

$$p_{ij}^k(t) = \frac{[\psi_{ij}(t)]^\alpha \times [\eta_{ij}(t)]^\beta \times [\eta\prime_{ij}(t)]^\gamma \times [\epsilon_{ij}(t)]^\delta}{\sum_{sl\in C(i)} [\psi_{ij}(t)]^\alpha \times [\eta_{ij}(t)]^\beta \times [\eta\prime_{ij}(t)]^\gamma \times [\epsilon_{ij}(t)]^\delta} \tag{2.42}$$

Where:

- $p_{ij}^k(t)$ and $\psi_{ij}(t)$ are transfer packet probability and pheromone metric respectively
- $\alpha, \beta, \gamma, \delta$ are the control parameters
- $\eta_{ij}(t)$ is the initial energy metric
- $\eta'_{ij}(t)$ is the second energy metric
- $\epsilon_{ij}(t)$ is the distance between node i and j

The initial energy metric calculated as bellow. The following Equation 2.43 the node with higher energy has a high probability in selection next-hop. Also, the calculation of the second energy metric follows Equation 2.44. The distance between nodes i and j is given by the Equation 2.45.

$$\eta_{ij}(t) = \frac{e_j(t)}{\sum_{sl \in C(i)} e_l(t)} \tag{2.43}$$

$$\eta'_{ij}(t) = \frac{(E - e_j(t))^{-1}}{\sum_{j \in C(i)} (E - e_j(t))^{-1}} \tag{2.44}$$

$$\epsilon_{ij}(t) = \frac{d_{jd}}{\sum_{sl \in C(i)} d_{ld}} \tag{2.45}$$

In this method, when ant k cannot find suitable next hop, it comes back to the previous node. Same as other algorithms, the ant deposits pheromone according to Equation 2.46.

$$\Delta\psi_{ij}^{k} = \frac{(hop_{max} - hop_{count_k} + \upsilon)^{1.5} \times E_{avg_k}}{hop_{count_k}} \tag{2.46}$$

Where:

- hop_{max} is the maximum allowed the number of hops
- hop_{count_k} is the number of hops for ant k
- E_{avg_k} the average energy of visited nodes

In LTAWSN, the energy is distributed among all nodes uniformly. The proposed method balances transmission besides decreases the energy consumption consequently increases the network lifetime.

2.3.3. Ant Colony Optimization in Combination with Hop Count Minimization (ACOHCM)

The ACOHCM is a routing algorithm that balanced energy consumption (Jiang & Zheng, 2018). This algorithm finds an optimal path with minimal energy consumption. This hybrid algorithm also tries to topology control that is inspired by the proposed method (Kiani et. al., 2015). In this approach, the authors proposed a dynamic energy threshold strategy different from the multipath approaches. Nevertheless, this study focused on a single source and destination point.

In the ACOHCM, at the first step hop counting the mechanism is applied. The hop count for the sink is 0. Other nodes hop count among neighbors plus 1. When the topology of the network changes, the hop counting mechanism is run again. So, the hop counts should be updated in time intervals. The second step is the energy threshold strategy for each node is applied. The initial energy of each node is C. The energy threshold is θ and the minimum threshold isθ_{min}. According to Equation 2.47, when an ant looking for next hop, a copy of the energy threshold θ is copied onθ_c. The next-hop node selected when current energy is greater thanθ_c.

$$\theta_c = \begin{cases} \lambda \times \theta_c\,, & \lambda\times\theta_c > \theta_{min} \\ \theta_{min}, & \lambda\times\theta_c > \theta_{min} \end{cases} \tag{2.47}$$

Where:

- λ indicates the extent of decrease of the threshold, $0 < \lambda < 1$

Then, the heuristic value is defined in Equation 2.48. The probability function of the ant k from node i to j to choose the next-hop node is in the following Equation 2.49.

$$\eta_{ij}(t) = \frac{E_j}{C}\xi_1 + \frac{d_{sum}}{d_{ij}}\xi_2 \tag{2.48}$$

Where:

- $\xi_1$and ξ_2 are the weighting factor where $\xi_1 + \xi_2 = 1$
- C is the initial energy of sensor nodes
- E_j indicates the remaining energy of node j
- d_{ij} is the distance between i and j
- d_{sum} refers to the sum of alternative distances between next nodes and the source node

$$p_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\left(\frac{E_j}{C}\xi_1+\frac{d_{sum}}{d_{ij}}\xi_2\right)^{\beta}}{\sum_{\mu\,\in allowed_k}\tau_{i\mu}^{\alpha}\left(\frac{E_\mu}{C}\xi_1+\frac{d_{sum}}{d_{i\mu}}\xi_2\right)^{\beta}} & j \in allowed_k \\ 0 & otherwise \end{cases} \tag{2.49}$$

Where:

- τ_{ij} is the pheromone value
- $\alpha\ and\ \beta$ are the relative influence of cumulative pheromone on the patch
- $allowed_k$ refers to the set of next hops

The pheromone update is important in the ACO algorithms, because each ant when in tour step leaves a chemical unit, so when a tour is finished the pheromone must be updated. In the ACOHCM, the pheromone update is given by equation 2.50.

$$\tau_{ij}(t+n) = \begin{cases} (1-p)\tau_{ij}(t) & L_{ij} \text{is not on the optimal path and} (1-p)\tau_{ij}(t) > \tau_{min} \\ & L_{ij} \text{is not on the optimal path and } \tau_{min} < (1-p)\tau_{ij}(t) \\ (1-p)\tau_{ij}(t) + \Delta\tau_{ij} & L_{ij} \text{is not on the optimal path and} (1-p)\tau_{ij}(t) + \Delta\tau_{ij} \geq \tau_{min} \\ \tau_{max} & L_{ij} \text{is not on the optimal path and} (1-p)\tau_{ij}(t) + \Delta\tau_{ij} \leq \tau_{min} \ or \\ \tau_{min} & L_{ij} \text{is not on the optimal path and} (1-p)\tau_{ij}(t) \leq \tau_{min} \end{cases} \quad (2.50)$$

Where

$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$ that m is the number of ants.

2.3.4. Improved ACO Approach

In this study, the authors proposed a routing algorithm for routes prediction in a hierarchy structure inspired by ACO and LEACH protocol (El Ghazi, et. al., 2014). However, this structure can cause overhead and energy costs in the network. In the following, we describe this method, in this method only they modify the probability function to ant k that moving from node s to node r Equation 2.51.

$$p_k(r,s) = \begin{cases} \frac{[\tau(r,s)]^{\alpha} \cdot [\eta(r,s)]^{\beta} \cdot [\delta(r,s)]^{\gamma}}{\sum_{r \in R_s} [\tau(r,s)]^{\alpha} \cdot [\eta(r,s)]^{\beta} \cdot [\delta(r,s)]^{\gamma}} & if\ k \notin tabu^{r} \\ 0 & otherwise \end{cases} \quad (2.51)$$

Where:

- $\tau(r,s)$ refers to pheromone value between r and s in Equation 2.52
- η is the heuristic value that here is initial energy
- R_s indicates the receivers node
- α and β are the control values
- $\delta(r,s)$ is a new heuristic value to control, it calculated in Equation 2.53
- $tabu^{r}$ is a list of packets received by node r

$$\tau(r,s) = \frac{(I-e_s)^{-1}}{\sum_{r \in R_s} (I-e_r)^{-1}} \quad (2.52)$$

Where:

- i is the initial energy of sensor nodes

- e_r is the residual energy of sensor node
- e_s is the residual energy of source node

And,

$$\delta(r,s) = \begin{cases} \frac{E_s}{\sum_{r \in R_s} E_r} & if\ Sink \in R_c \\ v & otherwise \end{cases} \tag{2.53}$$

Where:

- E_s is the residual energy of source node
- E_r is the residual energy of sensor node
- v is the constant value it changed base on the network environment

First, the heuristic value calculated by Equation 2.51, then a new heuristic value computed with Equation 2.52. This heuristic value indicated the remaining energy of sensor nodes. This new heuristic value gives a good result in energy consumption and reliability in packet delivery. After all, each node that chooses the next-hop using Equation probability function 2.50.

2.3.5. An energy-efficient ant-based routing algorithm (EEABR)

This method reduces the memory used in the sensor nodes to consider the energy quality of the paths found by the ants (Camilo et. al., 2006). They proposed a new ACO-based algorithm that sends a forward ant at regular intervals to determine the optimum route for each sensor node in the flat and location awareness architectures. EEABR was proposed for the decrease-consumed energy. On the other hand, it is a problem for the efficient consumption of resources because the time is not used well in this approach. The EEABR algorithm has the following Equations to update pheromone value, routing path and choose the destination.

$$\triangle T_k = \frac{1}{C-(Avg(E_k)-1/Min(E_k))} \tag{2.54}$$

Where:

- E_k is the vector carried by forwarding ant k
- C is the initial energy of nodes
- $Avg(E_k)$ refers to the average energy of the vector values
- $Min(E_k)$ is the minimum energy value of the vector

Equation 2.54 updates pheromone value that each ant k leaves in nodes visiting. The authors propose more improvements in equation 2.54 to reduce energy consumption again. They propose Equation 2.55 to update pheromone.

$$\triangle T_k = \frac{1}{C - \left[{}^{EMin_k - Fd_k}\!/_{EAvg_k - Fd_k} \right]} \tag{2.55}$$

Where:

- Fd_k is the number of nodes that forward ant k has visited

After the pheromone update, the probability function is calculated as Equation 2.56.

$$T_k(r,s) = (1-p) \cdot T_k(r,s) + \left[{}^{\triangle T_k}\!/_{\varphi \cdot Bd_k} \right] \tag{2.56}$$

Where:

- φ is a coefficient
- Bd_k is the traveled distance

Generally, the goal of the EEABR algorithm is to reduce the size of ants in among of network communication. In this way, each node keeps only neighbors in the sink direction. Therefore, the number of ants decreases but does not decrease the energy consumption for data transmission.

Other studies inspired by the ACO algorithm that almost improve network lifetime. In this section, we describe some of them briefly. Most of the proposed methods have modifications to probabilistic function. As a result, we highlighted the improvements element in the following.

2.3.6. Energy Optimization of Ant Colony Algorithm in Wireless Sensor Network (IACAEO)

The authors proposed an energy-efficient method for WSNs based on ACO (Li et. al., 2017). They allow each node in the network to save the distance and remained energy of neighbors. In this study, the authors aimed to prevent too much energy consumption of a particular local node by proposing an improved ant colony algorithm. Ants choose the path due to low pheromone concentration and high-energy nodes in each path. However, it uses very complex functions in both the path selection and the phenomenon update processes, so the functionality of the algorithm is being discussed in real platforms.

2.3.7. An Energy Aware Routing Based on Swarm Intelligence (ARNet)

The ARNet is a new ACO algorithm for the path-finding mechanism with minimum cost in the network (Lv et. al., 2014). Each ant obtains the path length with the number of nodes in the path. This study was focused on a single source and destination point similar to many other types of research based on other ACO algorithms. Furthermore, it also does not use memory efficiently.

2.3.8. Efficient quality-of-service support in mobile opportunistic networks (QoS-aware)

In this study, the authors introduced the mobile opportunistic self-organization network that does not require a complete path between the source node and the destination node (Liu et. al., 2014). However, the proposed algorithm does not focus on and discuss the performance of their method whenever the source and destination points are more than one. It is also not good in memory consumption.

2.3.9. A Nature-Inspired Data Gathering Protocol (T-ANT)

This algorithm is an energy-saving algorithm that has been proposed in the same hierarchical structure based on a homogeneous cluster head (CH) distribution (Selvakennedy et. al., 2006). In the election of CHs, a herd of ants is used where an ant corresponds to a control message. The ant goes as deeply as the network is bounded by its time-to-live (TTL) field. However, in this method, over the specific nodes selected is more traffic and it causes that they are failed or discharged faster than the other nodes. Indeed, it is not a balanced energy depletion.

2.3.10. An energy-balanced ant-based routing protocol (EBAB)

The EBAB divides the network into unequal sizes of clusters in order to balance energy consumption (Wang et. al., 2009). Clusters closer to the BS have smaller cluster sizes in order to maximize the lifetime of the network so the authors inspired by the proposed method (Kiani et. al., 2013-b). It is more focus on the inter-cluster communication and the optimized CH election has been not discussed. Each CH node tries to access the base station using the other CHs. For this, a suitable ACO algorithm has been proposed.

2.3.11. Multipath Routing Protocol based on Clustering and Ant Colony Optimization (MRP)

The MRP introduced a multipath routing protocol for clustering based on ACO (Yang et. al., 2010). With the forward and backward ants, the route is taken from each CH node to the BS node but the same path will be used until the energy level of the corresponding CH node is below 50%. Therefore, the energy depletion of other nodes on the selected path is expensive and will cause unbalanced energy consumption.

2.3.12. An ant colony clustering routing algorithm (ACA)

In this method, the authors focused on CH selection. There are two parameters for CH selection; remaining energy and the distance between the node and cluster heads (Wang et. al., 2009).

The main goal of the ACA is to find the optimal path between CH and sink/BS. As this method benefits the ACO algorithm, the pheromone value has an impact on path selection. The ants settled on each CH and start exploring the new path, besides the generated path is stored on a matrix. Then, the path with the minimum cost is chosen as the optimal path. The network lifetime of the ACA has a good performance on hierarchical-based networks.

CHAPTER THREE: PROPOSED METHODS

- A new energy-efficient clustering method in hierarchical based on wireless sensor networks
- Energy Efficiency in the Heterogeneous Sensor Networks (EEHRSN)
- Improved version of HEEL algorithm based on metaheuristic algorithms
- Routing algorithm based on metaheuristic algorithms
- Concurrent path finding in real-time wireless sensor networks by a new routing protocol inspired by ant colony optimization

3.1. A New Energy-Efficient Clustering Method based on Hierarchical Wireless Sensor Networks (HEEL)

3.1.1. Motivation and Challenges

Since the main aims of this thesis to contribute to the energy efficiency of the wireless sensor networks, in this section, we have described a novel way to clustering methods in hierarchical networks. As mentioned in the introduction section wireless sensor networks, suffer from limited energy supply. There are some effective approaches to overcome this fundamental problem. In clustering methods, the important challenge is cluster head selection. The main responsibilities of the cluster heads are data collection and aggregation. Data aggregation done at some selected nodes can reduce energy consumption. Another approach to reducing energy consumption is to put some nodes in sleep modes when they are inactive to save energy (Dogan, 2016). For example, in target tracking applications each sensor node is necessary to be aware of every local event so they have to be awake all the time. Sensor nodes have limited energy supply.

In a hierarchical structure, the network is divided into clusters. Each cluster has a cluster head and cluster member. Unlike networks based on flat architecture, in hierarchical networks cluster heads directly can communicate with the base station. Cluster members communicate only with corresponding cluster heads. In hierarchical routing protocols, data aggregation decrease communication between a sensor node and base station. Data aggregation reduces the transmitted data to base station.

In the previous proposed methods like the LEACH (Heinzelman et. al., 2000) the cluster head selects as in random way. In section 2.2 it is widely explained. Also there are many various version of the LEACH algorithm that improved the performance of this algorithm. There are different cluster head selection algorithms to increase network lifetime; LEACH (Heinzelman et. al., 2000), MODLEACH (Mahmood et. al., 2013), HEED (Younis & Fahmy, 2004), and LEACH-B (Tong & Tang, 2010).

In HEEL we consider residual energy, distance between a sensor node and the base station and number of links to neighbors. Network lifetime strongly depends on each node's energy level. The proposed methods supposed the network with same features it means the network sensor nodes are in homogenous type. In homogeneous networks all feature of sensor are same like energy level, transmission range.

HEEL based on energy level of each sensor and neighbors too. The number of neighbors is important in CH selection. The region with a few amount of sensor is need one or few cluster heads with this point that cover the region. Also in the HEEL algorithm, we focused on overhead of sensor nodes. Sensor nodes that chosen as CH previous has rule in cluster head selection in that round unlike LEAH algorithm.

While in this network structure, the sink/BS is on stable location. In most of clustering methods the near node to the sink/BS drain own energy earlier than others. Since there are, near to sink/BS and have more connectivity with sink/BS. If the near sensor nodes dead earlier than other, remain sensor nodes should transfer data packet is high transmission range in this way consume more energy. We consider the hop size to Sink/BS. In the proposed method, we involved the hop size to cluster head selection.

The routing phase of this method is based on the LEACH algorithm routing protocol. We have not more enhancement in the routing phase. We just focused on cluster head selection. As in simulation phase of the HEEL, we have come to some situations to choose this routing protocol that we explain this problem in the next section. Also in the remaining parts of this section of the HEEL algorithm, we shows the pseudo code and the flowchart. In addition the simulation result of the HEEL in comparison of the some algorithms will show in the result and conclusion section.

3.1.2. Proposed method

Let us consider a network in which the node is deployed randomly. For instance, let us consider a node n_i, where $1 \leq i \leq n$ that is located far away from the base station. Also, assume that the base station is located at the center of the network. For this specific case, transferring the collected data to the base station requires more energy and increases the cost. As the node n_i does not have enough energy to send the data, it can be said that the data will not reach to the base station. When n_i is selected as the cluster head, it will not have enough energy for the transmission with respect to the base station. Under these assumptions, we have proposed the HEEL in order to select the cluster head.

Our motivation for this work is to develop a new clustering method. Our network model will be based on the hierarchical network model that contains clusters, cluster members, cluster heads and a base station. The succeeding text enumerates the description of our architecture

- Cluster: Consist of a set of sensor nodes that have a transmission with each other.
- Cluster member: In each cluster, there are sensor nodes that have the duty of gathering data and acting with respect to the events.
- Cluster head: Each cluster has one cluster head that is responsible for collecting data from cluster members, aggregating the data and transmitting it to the base station or to the higher-level cluster head.
- Base station: It is located at the center of the network on top of the network hierarchy. It collects data from the cluster heads. Also, it has an unlimited energy supply.

The power model used is similar to the HEEL algorithm (Heinzelman et. al., 2000). This model is proposed in Equations 3.1 and 3.2. These equations calculate the amount of energy consumed during transmission and receiving. The power dissipated while sending k bits of data can be calculated as:

$$E_{tx}(k,d) = \begin{cases} E_{elec} \times k + e_{fs} \times k \times d^2 & ,d<d_0 \\ E_{elec} \times k + e_{mp} \times k \times d^4 & ,d\geq d_0 \end{cases} \tag{3.1}$$

And to receive k bit of data radio expends:

$$E_{rx}(k) = k \times E_{elec} \tag{3.2}$$

Where:

- E_{elec} is the amount of energy consumption in the transmitter or receiver
- e_{mp} is multipath fading energy consumption
- e_{fs} is free space energy consumption and d is the distance between the nodes

HEEL is based on clustering and cluster selection in wireless sensor networks. This algorithm use application layer of a wireless sensor network. The purpose of the invention is to reduce the number of dead nodes, and increase the lifetime of the network. Also, another goal of this algorithm is to optimize the energy consumption throughout the entire network.

As LEACH algorithm is based on clustering, it has some problems in selecting cluster header. In this paper, we propose a different method for choosing a cluster head. For each node we propose 4 parameters that each of these parameters saved in a table that named neighboring table in each node. Neighboring table shown in Table 3.1.

- Node energy
- Energy of node's neighbors
- Number of hops
- Number of links to neighbors
- Cluster ID

Table 3. 1 Neighbor table

Node ID	Node Energy	Energy of node's neighbors	Number of hops	Number of links to neighbors	Cluster ID
i	E(i)	Es(i)	H(i)	L(i)	Cl(i)

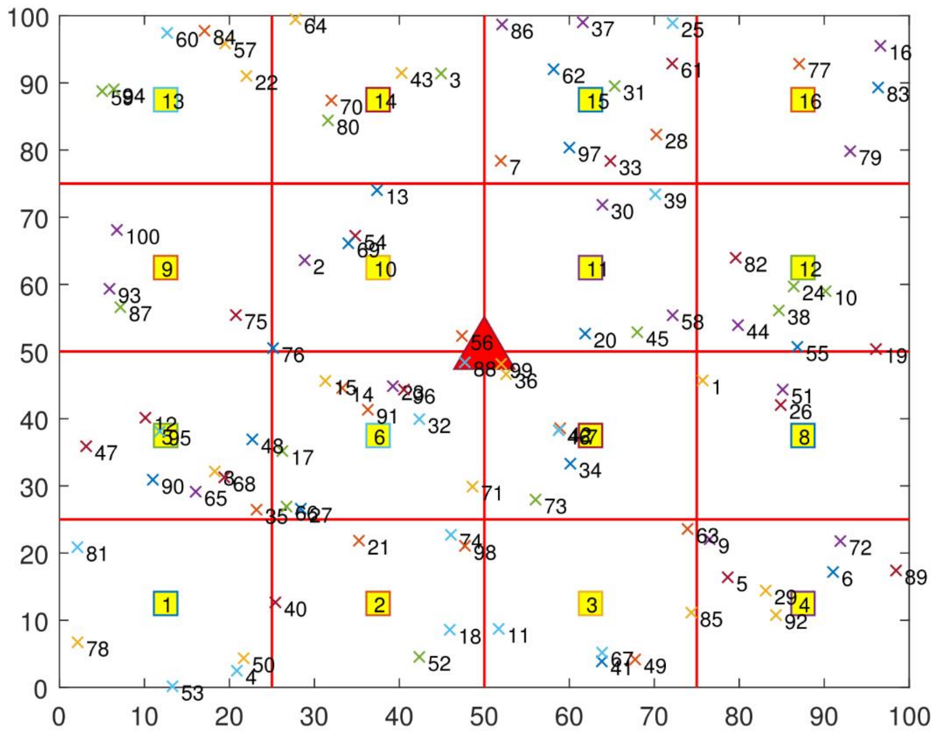


Figure 3. 1 Network Structure

Source Address	Get Node Energy	CRC

Figure 3. 2 Request data packet

Source Address	Destination Address	Node Energy	CRC

Figure 3. 3 Answer data packet

After deployment, each node neighboring table will be created. In this step, each node broadcasts a packet to aggregate data from neighbors and save it in the table. Each node broadcast a packet as shown in 3.2 to get neighbor node energy and find neighbors in the step is based on the Received signal strength indicator (RSSI). After this neighbor node broadcasts a message like as shown in 3.3 to the source node. In this situation, node *i* update own neighbor table. Calculate number of links and sum of neighbor nodes energy. In neighbor table, number of hops to base station is done by a message that broadcast by base station to each node. Also each node will define own cluster and then save that cluster ID in neighboring table.

Node Energy: Energy of a node is a crucial element in wireless sensor networks. Residual energy of the node can help us while choosing the cluster head; therefore, it is very effective. If the residual energy of a node is less than the required energy for the transmission of the data, choosing these nodes for a cluster is useless. One of the important tasks of the cluster is to collect information from cluster members. Cluster head sends aggregated data to base station. If the residual energy is less than a threshold value, then the node will not be selected as the cluster head. The residual energy of a node varies per round depending on the task of the cluster member or cluster head.

Energy of Node's Neighbors: For the selection of cluster head, we also consider the energy of node’s neighbors. The energy of the neighbor of the candidate cluster head must be more than the amount of necessary energy for data transmission. Otherwise, the selection of this cluster head will not be useful. In this case, neighbors will be dying and the data will not be transmitted. In addition, the distance between cluster head and the base station will have an effect on the selection of the cluster. Each node count number of their neighbors from received message packets as shown in Figure 3.3.

Number of Hops: Consumption energy of a node is also important in cluster head selection. It is related to the distance between each of the nodes. The candidate cluster head is not located far from the base station, because the dissipation energy for each distance will be different. HEEL tries to select a node that is closer to the base station, as it will save more energy. Therefore, we consider the number of hops to base station in selection cluster head method. In this step, BS sent a packet to each sensor node to calculate the distance between BS and sensor nodes.

Number of Links to Neighbors: The next one is the number of links to neighbors, in other words, the number of paths between a node and the other nodes. Here, for each node, a path table is considered. After finding the number of neighbors of a node, the information of each node is stored in the neighbor table. The purpose of finding number of links to neighbors is to select a node that has a greater number of neighbors because it covers a large number of nodes. This will cause the selection of a node as cluster head that has maximum number of neighbors. In this case, data aggregation will be done and there will be no clusters without the nodes.

As mentioned above, we consider four parameters to select the cluster head in each cluster. A specific value is calculated for each node by using the following Equation 3.3 to select cluster head:

$$Fitness(i) = \left(a_1 * E(i)\right) + \left(a_2 * L(i)\right) + \left(a_3 * Es(i)\right) + \left(a_4 * H(i)\right) \qquad (3.3)$$

Where:

- $E(i)$ is residual energy of node i
- $L(i)$ is number of links of node i to its neighbors
- $Es(i)$ is Energy of node i's neighbors
- $H(i)$ is number of node i's hops to the base station

Number of hops to the base station and number of links to the neighbors have constant values during the lifetime of the network. So, we have used fixed coefficients in Equation 3.3. In addition, Node’s energy and energy of node’s neighbors have different and more effective coefficients. Here,a_1, a_2 , a_3, and $a_4$are weights that are assigned to each of the parameters. These weights are related to the usage of the application and purpose of the network. For example, if in the network our goal is to have a greater energy efficiency, it is necessary to use $a_1$and $a_3$with a higher value. As another example, if the distance is the most important component for the network, a_{41}must have a higher value to use low power energy. In Equation 3.3,$a_1 + a_2 + a_3 + a_4 = 1$, where $a_i \in (0,1)$.

The obtained value from Equation 3.3 for each senor node of a cluster is compared to other sensor nodes value. A sensor node will be selected as cluster head that has the maximum value in that cluster. Cluster head broadcasts a package that contains cluster head's ID. After all, in each round the network has sixteen cluster heads. CH in neighbor table has own neighbors, so in the broadcasting package, the ID of neighboring sensor node is shared.

So cluster members aware from their cluster head and communicate with that. As in LEACH algorithm in HEEL algorithm for inter cluster communication TDMA schedule used to avoid collision. Intra cluster head communication is done in low energy amplification. CH's aggregate collected data from cluster members, so they must forward data to BS. HEEL algorithm followed a hierarchical structure, so each CH must forward data to up-warding layer near to BS. In the following the pseudo code (Figure 3.4) and flowchart (Figure 3.5) of the HEEL shown.

HEEL's algorithm

Input: All of nodes in each round
Output: Selected cluster head
N=Total Number of nodes
J=Max cluster numbers
E(i)=Node energy
Es(i)=Energy of node's neighbors
L(i)=Number of links to neighbors
H(i)=Number of hops to base station
Rmax= Maximum round number
For r=1:Rmax do
If j<=J then
j= cluster members size
for i=1:j do
if E(i)>0 then
Fitness(i)=(a1*E(i))+ (a2*L(i))+(a3*Es(i))+ (a4*H(i))
else
Fitness(i)=0;
end
find max Fitness
Broadcast selected CH-id to Members
State(i)=CH
Update(E(i))   //Calculate Dissipation Energy
end
j ++
end
r ++
end

Figure 3. 4 Pseudocode of the HEEL algorithm

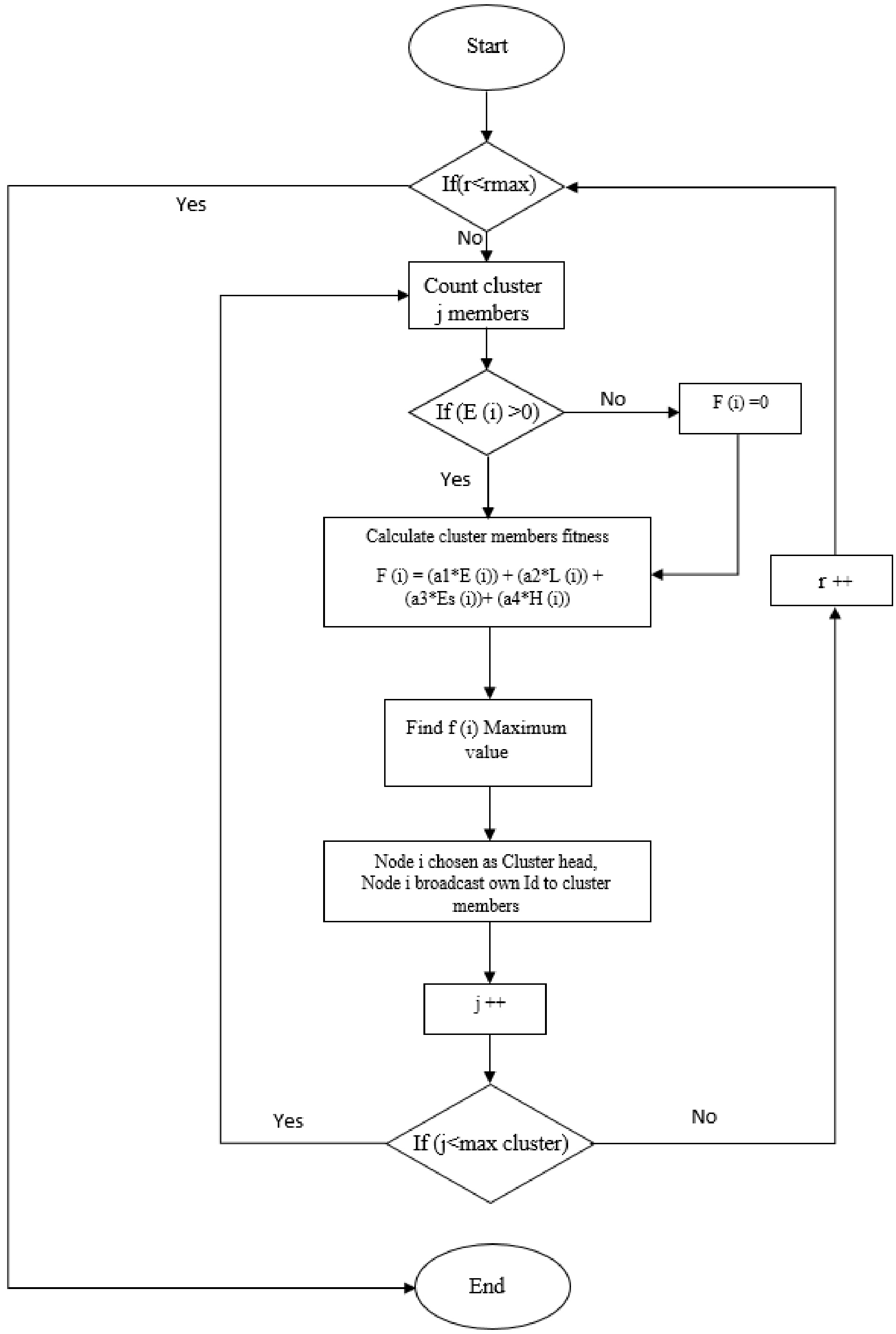


Figure 3. 5 Flowchart of the HEEL algorithm

3.1.3. Configuration Parameters for Simulation

We have simulated our method on MATLAB. The HEEL and comparison algorithm in the same conditions simulated in this software. The performance metrics such as the throughput, total residual energy, and the number of alive nodes. The algorithms LEACH (Heinzelman et. al., 2000), NR-LEACH (Al-Baz & El-Sayed, 2018), MODLEAD (Mahmood et. al., 2013), LEACH-B (Tong & Tang, 2010), PEGASIS (Lindsey & Raghavendra, 2002), EACLE (Yanagihara et. al., 2007), and HEED (Younis & Fahmy, 2004) are considered to compare the performance metrics of the algorithms.

The input parameters are outlined in Table 3.2. The initial energy of each node, data packet size, radio, sensing radius, and sensor energy consumption are the values that are the same and static for all algorithms in the simulation phase. The simulation capable of modifying the specifications of sensor nodes. To evaluate the performance of the HEEL we consider two different scenarios.

Table 3. 2 Configuration parameters of the HEEL algorithm

Parameters	Values
Network Size	100*100 m
Node Size	100
Base station location in scenario 1	50*50 m
Base station location in scenario 2	0*50 m
e_0	0.1 J
Data packet size	4000 bits
e_{fs}	10 pj/bit/m2
e_{mp}	0.0013 pj/bit/m4
e_{elec}	50 n J/bit

Where:

- e_0 indicates the initial energy of each node
- e_{fs} is free space energy consumption
- e_{mp} is the multipath energy fading
- e_{elec} is the amount of energy consumption in send and receiver

In the network model, all of the sensor nodes are deployed randomly and in two-dimensional space. In the HEEL, all sensors are static and have a fixed position.

Also, the sink/BS has unlimited energy supply. Two scenarios are considered as follows:

- **Scenario 1**: a network in a 100×100 meter field and the base station is located in the center of the network50×50.
- **Scenario 2**: a network in a 100×100 meter field and the base station is located on the left edge of the network0×50.

We assume that 100 number of sensors are dispersed on 100×100 (meters) operational area. Below are properties in our network:

- The nodes are deployed randomly in a specified area.
- The base station is located at a fixed point in the center of the network (scenario 1).
- The base station is located on the left side of the network (scenario 2).
- The whole network divided into 16 cluster
- Each cluster will have one cluster head
- All nodes are homogeneous, initial energy is fixed in each node.
- Each node can send data directly to the base station.
- All nodes have the same transmission range.
- All nodes have a unique identification number.
- Nodes are not mobile.

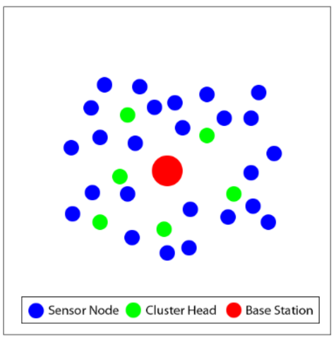


Figure 3. 6 Network architecture

The network range size in meters is100×100, the base station is in the first scenario in the center of the network at 50×50 coordinates. In the second scenario, the base station is located on the left side of the network. The initial energy of all nodes is fixed at the beginning of the network and it is set to 0.5 J. Assuming that all nodes are homogeneous, data packet size per round is 4000 bytes. We suppose that all of the sensor nodes have at least one neighbor.

3.1.4. Comparison and Results

This section presents the simulation results of the HEEL algorithm. As mentioned before we have two scenarios to evaluate the performance of the proposed method in two different base station positions. All of the comparison algorithms simulated with the same parameter configuration as Table 3.2. Also, the same performance metric like, total residual energy, throughput and the number of alive nodes are the same outputs for all algorithms.

Total residual energy: This metric parameter describes the remaining energy in the network. In most of the comparison metric, the value explained in the different number of rounds. According Figures 3.7 and 3.8, it clearly indicates that the HEEL algorithms has good performance in energy consumption in the whole network. Also the Table 3.3 presents the amount of total energy in the 10 point of 500 round for scenario 1.

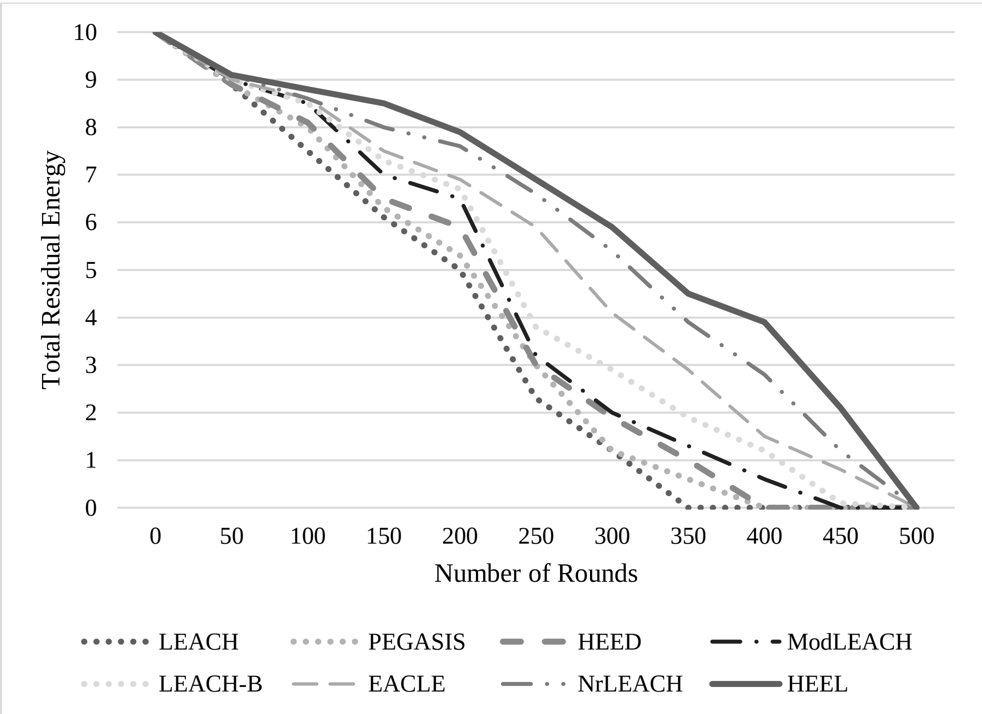


Figure 3. 7 Total residual energy in scenario 1

Table 3. 3 Total residual energy in each iteration in scenario 1

Round	LEACH	PEGASIS	HEED	MODLEACH	LEACH-B	EACLE	NrLEACH	HEEL
0	10	10	10	10	10	10	10	**10**
50	8.9	8.9	8.9	9	9	9	9.1	**9.1**
100	7.5	8	8.1	8.5	8.5	8.6	8.6	**8.8**
150	6.11	6.3	6.5	7	7.3	7.5	8	**8.5**
200	5	5.3	5.9	6.5	6.7	6.9	7.6	**7.9**
250	2.7	3	3	3.2	3.8	5.9	6.6	**6.9**
300	0	1.2	1.9	2	2.9	4.1	5.4	**5.9**
350	0	0	0	1.3	1.9	2.9	3.9	**4.5**
400	0	0	0	0	1.2	1.5	2.8	**3.9**
450	0	0	0	0	0	0	1.2	**2.1**
500	0	0	0	0	0	0	0	**0**

The total residual energy in scenario 2 is different with scenario1. Since the location of the sink/BS is different. In this way, the cluster head transmission with sink/BS is more than scenario 1.

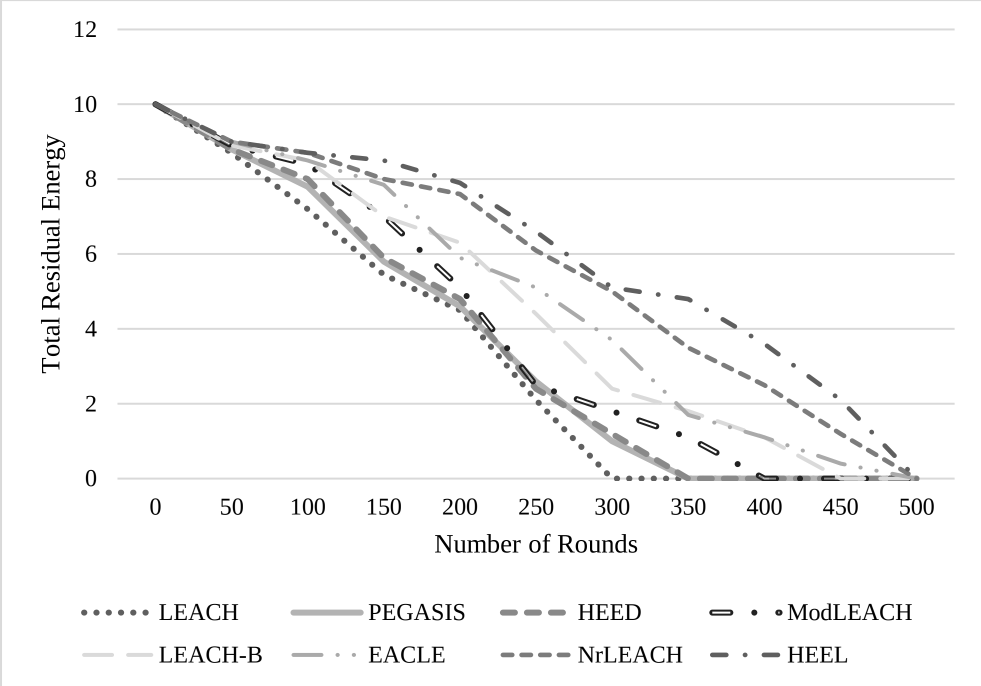


Figure 3. 8 Total residual energy in scenario 2

Table 3. 4 Total residual energy for each iteration in scenario 2

Round	LEACH	PEGASIS	HEED	ModLEACH	LEACH-B	EACLE	NrLEACH	HEEL
0	10	10	10	10	10	10	10	**10**
50	8.7	8.8	8.8	8.9	8.9	9	9	**9**
100	7.2	7.8	8	8.4	8.5	8.5	8.7	**8.7**
150	5.45	5.8	5.9	7	7	7.85	8	**8.5**
200	4.5	4.6	4.8	5.1	6.3	5.9	7.6	**7.9**
250	2.1	2.6	2.4	2.5	4.4	5.1	6.1	**6.6**
300	0	1	1.2	1.8	2.4	3.7	5	**5.1**
350	0	0	0	1.1	1.8	1.7	3.5	**4.8**
400	0	0	0	0	1.1	1.1	2.5	**3.6**
450	0	0	0	0	0	0.4	1.2	**2.1**
500	0	0	0	0	0	0	0	**0**

Throughput (Thr): This is a performance metric used to evaluate the performance of the network. Throughput is the number of data per unit of time that is sent by the sensors to the base station or cluster heads. This is measured in bits per second. Generally, throughout is the data packets per second. It can be shown like bits/s or bps. It can be given by Equation 3.4.

$$thr = \frac{transfer\ packets * packet\ size}{total\ time} \tag{3.4}$$

Figure 3.9 that shows throughput from cluster heads to sink/BS results on the HEEL algorithm and other algorithms for scenario 1. As shown in bellow, the throughput value for each of these algorithms until the half the network round is like equal. The results observation in throughput shows that HEEL algorithm improved the throughout about 54% compared to the ELACH. Also about others algorithms the HEEL performing better results.

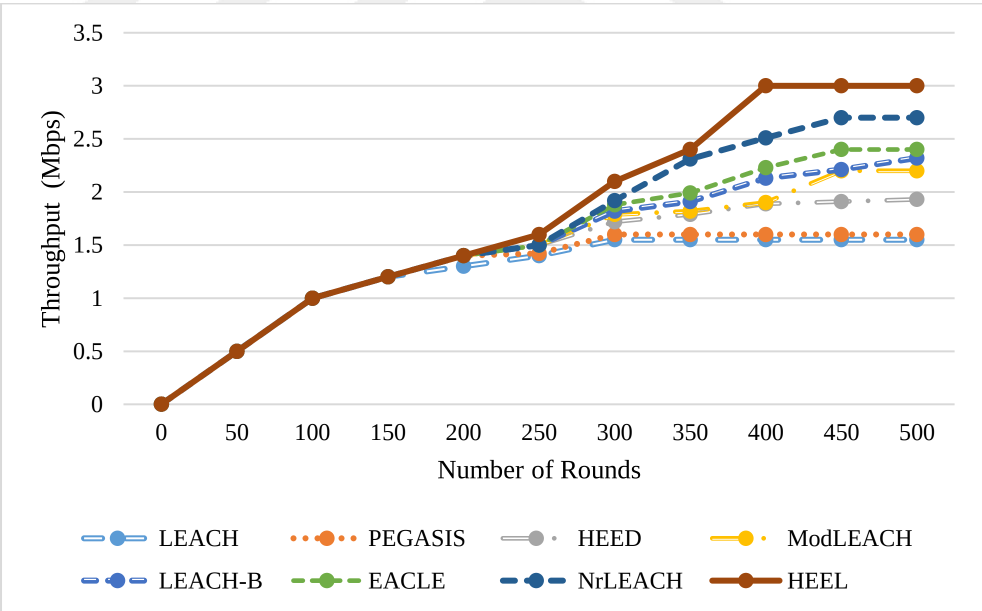


Figure 3. 9 Throughput of HEEL algorithm in scenario 1

In addition, the throughput value from cluster members to cluster heads shows that the HEEL algorithm has better performance too. Since, in the HEEL the cluster head selection base on four parameters that chose the optimal cluster head. CH in most of time in clusters has good performance. Also, the communication between CHs and cluster members is available until the network total energy became a drain.

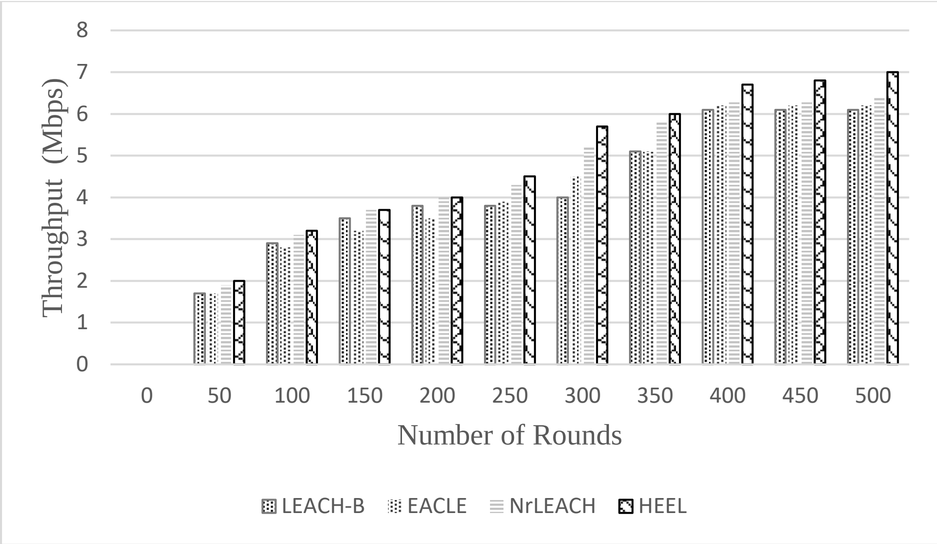


Figure 3. 10 Throughput value for LEACH-B, EACLE, Nr-LEACH and HEEL in scenario 1

As mentioned before, in scenario 2, the sink/BS located in the left edge of the network in0×50. Figure 3.11 and 3.12 presents the throughput in the second scenario. As shown in figures, when the sink/BS located on the left side of the network, the total number of data packets in increased in comparison of the sink/BS located in the center of the network.

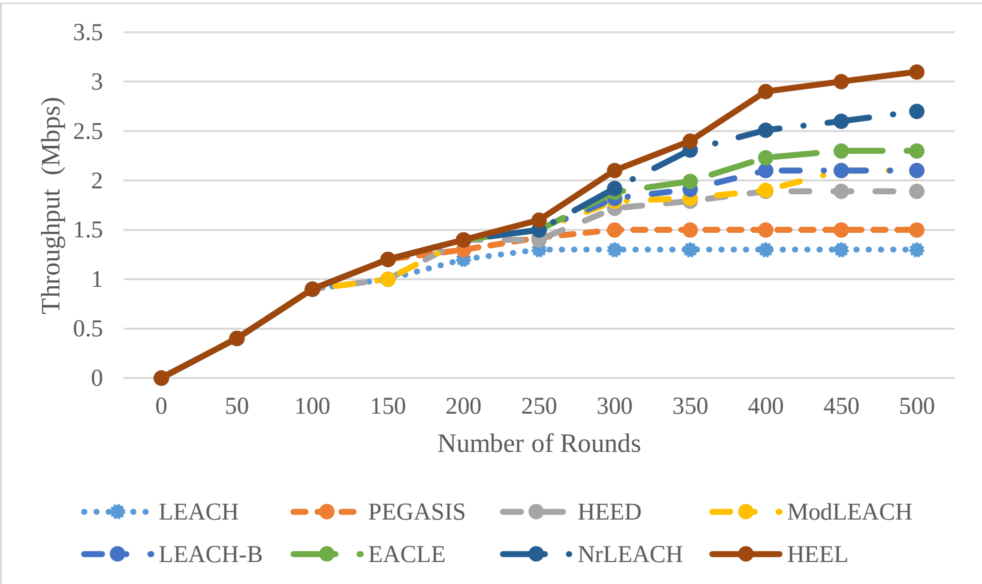


Figure 3. 11 Throughput of HEEL algorithm in scenario 2

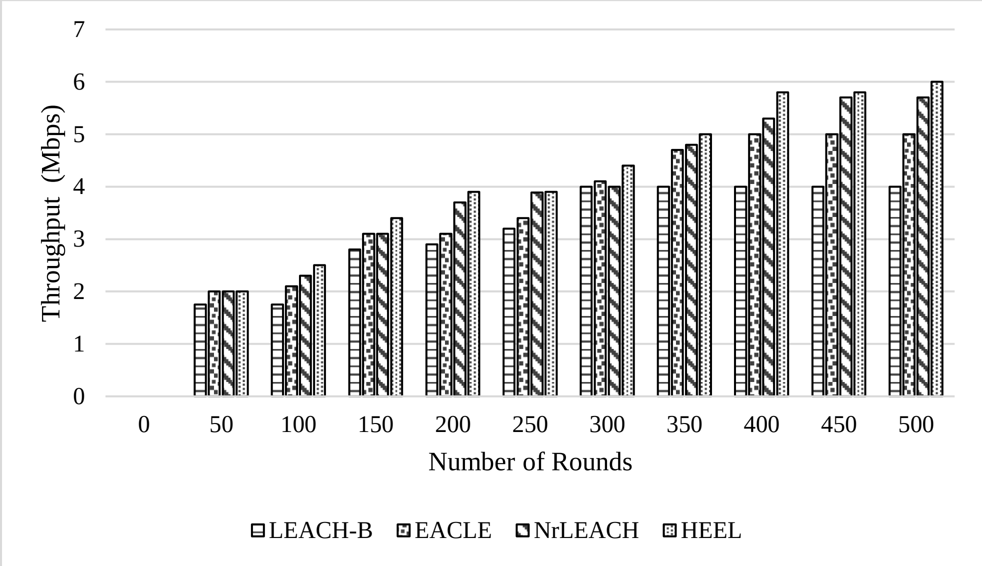


Figure 3. 12 Throughput value for LEACH-B, EACLE, Nr-LEACH and HEEL in scenario 2

The number of alive nodes: This metric indicates the whole network alive sensor nodes after the rounds finished. In general, the relationship between the number of alive and dead nodes is vice-versa. According Figure 3.13, the proposed method HEEL has the more alive nodes in comparison of other algorithm in the whole network in scenario 1. For example, the LEACH the last node dead in the 320_{th} round of the network, while the last node dead in other algorithms like PEGASIS, HEED, MODLEACH, LEACH-B, EACLE and Nr-LEACH later than LEACH.

Table 3. 5 Number of alive nodes in scenario1

	LEACH	PEGASIS	HEED	ModLEACH	LEACH-B	EACLE	NrLEACH	HEEL
Last node dead	320	390	395	405	440	450	455	485

According Table 3.5, the HEEL algorithm last node dead in the 485_{th} round. In other means, after this round there are not any alive node to transmission. LEACH algorithm the last node dead in the 320_{th} round.

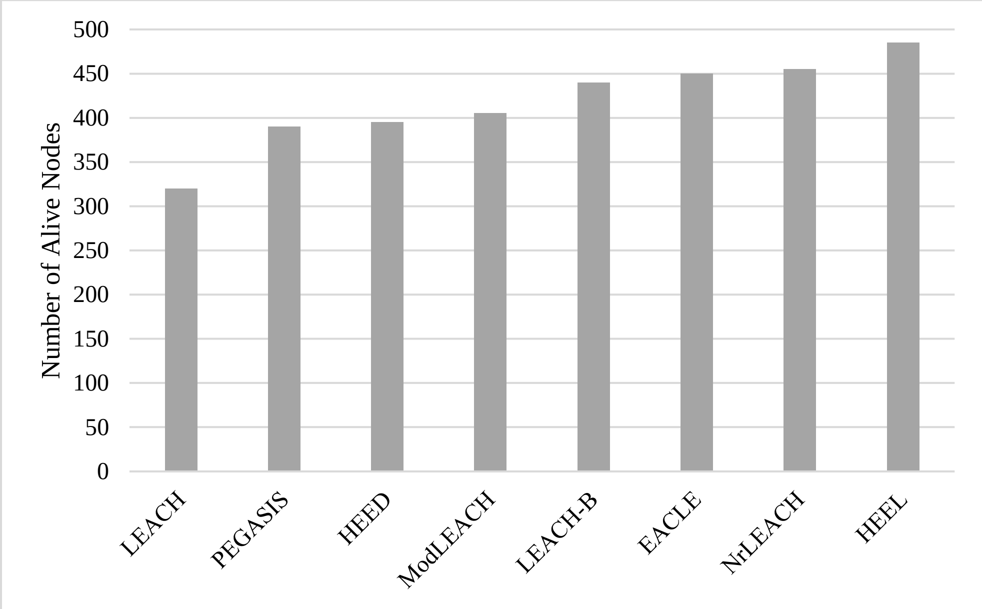


Figure 3. 13 Number of alive nodes for each algorithm

In scenario due to the sink/BS is located on the left side of the network, the number of alive and dead nodes is different. Due to the node that near to the sink/BS consume more energy than others in transmission. Because the far cluster heads should be, transfer the data packets with near cluster heads to the sink/BS. As a result, the number of alive nodes in scenario 2 is lower than scenario 1.

Table 3. 6 Number of alive nodes in scenario 2

	LEACH	PEGASIS	HEED	ModLEACH	LEACH-B	EACLE	NrLEACH	HEEL
Last node dead	284	371	374	387	415	435	440	470

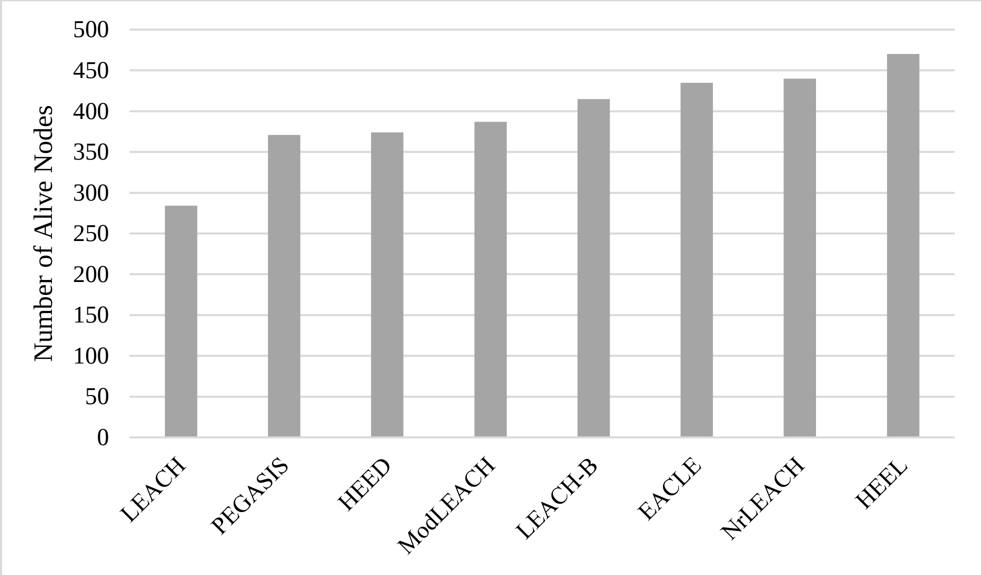


Figure 3. 14 Number of alive nodes for each algorithm

3.1.5. Conclusion

One of the main problems in the WSNs is the energy consumption. Energy efficiency is highly critical issue in WSN. Most of studies tries to balance between the resources. In most of applications of the wireless sensor networks, the end-user can't change battery also this energy supply is not rechargeable. As a result, the researchers try to have improvement in the application layer of the WSN. The popular way in application layer is enhancement in the energy efficient routing algorithms, also in the network topology.

Hierarchical based network topology is used in most of the WSN application. Since, this type of network can be applied in most of the application and also many approaches to enhance the energy efficiency. Clustering schemes in popular in the hierarchical based networks. Clustering methods try to enhance the network lifetime besides reduce overhead.

As in the clustering methods, the main goal is to increase network lifetime. The number of alive node has the direct relation with network lifetime. Until now most of researchers focused in this problem to propose novel methods. As the introduction of this section mentioned, the state-of-are algorithm is LEACH that there are many variants of this.

Clustering methods like ELACH, consist of clusters and cluster heads. The main operation is selection the cluster head. In the LEACH algorithm the cluster head selection process in random state. In this case, the number of nodes that chosen as CH and others not important, since the selection state is random. In variants, versions of the LEACH authors add some parameters in the cluster head selection like, residual energy.

The purpose of the current study was to determine several parameters that have a positive impact on the cluster head selection. HEEL proposes a new method for selecting cluster head that improves the network lifetime and throughput. HEEL is based on node's energy and location of node. Node energy, energy of node's neighbors, number of hops, number of links to neighbors are parameters that have important effects on the cluster head selection. The results show that HEEL provides energy efficiency in wireless sensor networks.

It can be used in various case studies. For example, in different applications of wireless sensor networks such as healthcare, monitoring and agriculture. Moreover, it is a good suggestion for IoT-based projects. In this method, the life of the system is long as well as efficient and balanced use of resources is another advantage.

As a future work, it is planned that the parameters in the presented function will be obtained by a metaheuristic algorithm and machine learning methods, not by classical simulation and experiment methods.

3.2. Energy Efficiency in the Heterogeneous Sensor Networks (EERHSN)

3.2.1. Introduction and related works

Wireless sensor network is a sub category of ad-hoc wireless sensor networks (He et. al., 2003). Mainly the wireless sensor network divided into two categories: homogenous sensor network and heterogeneous sensor network. Generally, heterogeneous wireless sensor networks (HWSN) have differences in sensor nodes features. In this way, there are special energy-efficient protocols in HTWSN. Also, researchers try to improve network lifetime use node heterogeneity (Tanwar et. al., 2015). Consider a network that sensor nodes should be transmitting data packets in a large distance, so they required high energy. In this way, the sensor nodes drain energy supply early. This is impossible to change the battery, due to most of the sensor networks deployed in harsh environments.

Besides, the wireless sensor network is a kind of distributed system that works cooperatively; a crashed sensor node has an impact on whole network performance. The proposed algorithms should be robust in the face of these issues. On the other hand, heterogeneous sensor node network can have different sensor nodes with high and low battery supply. Data transmission and collection are the same as the homogenous wireless sensor network. As the same flat and hierarchical based topologies are well known in HWSN (Rostami et. al., 2018).

In heterogeneous wireless sensor networks, the sensor nodes have a difference in features such as energy, computation, communication, etc. The main advantages of the HWSN are reducing the latency and increase throughput (Tanwar et. al., 2015). Since in heterogeneous sensor nodes, the communication between the sensor nodes is less than homogenous sensor nodes. As mentioned before the energy efficiency is a challenge in WSN. An interesting advantage of HWSN is its energy heterogeneity. Besides, strong computational processing can provide superior heterogeneous sensor nodes. Furthermore, the long and small distance for communication is not a challenge in HWSN. As node heterogeneities, energy heterogeneity is more important because both computational and communication heterogeneity will consume more energy. Heterogeneity has affected in wireless sensor networks such as network lifetime, delay, reliability.

- **Network lifetime:** the consumed energy in transmitting data packets from sensor nodes to sink/BS is lower than other networks. Generally, the network lifetime will be increased.
- **Delay:** Computational heterogeneity can reduce processing delay time, and communication heterogeneity can reduce waiting time, thus reducing response time.
- **Reliability:** mostly, in the sensor networks, connections tend to be less stable, and each hop significantly reduces end-to-end transmission rates. With heterogeneous nodes, there will be fewer hops between the normal sensor nodes and the sink/BS. Therefore, a heterogeneous sensor framework can achieve a higher end-to-end transfer rate than the homogeneous sensor network.

Mainly, heterogeneity in the HWSN is in two and three-level. Consider, we have *n* nodes in the network where *m* of these nodes have nodes with *a* value higher energy order than others. So, this network is a two-level heterogeneous network. In the related section of the thesis, we explained state-of-art methods in the HWSN such as; SEP (Smaragdakis et. al., 2004), DEEC (Qing et. al., 2006), EECH (Kumar et. al., 2009a).

In the proposed method, we consider a three-level heterogeneity of network. Each sensor is stable and the sink/BS has unlimited energy supply. Generally, the network is divided into 16 virtual grids. The proposed method has 20 static cluster head. Three types of difference sensor nodes have a difference in energy and communication. Normal, advance and super nodes are the sensor nodes that consist of the network. The cluster head is selected statically. The proposed method has a routing algorithm to transfer collected data from clusters to transfer sink/BS. The proposed method in comparison to other algorithms has good results in network lifetime and data packet delivery. The following section explained the description of the proposed method.

3.2.2. EEHRSN Method

Hierarchical based networks have better performance in energy consumption than flat-based networks. Heterogeneous wireless sensor networks extend network lifetime due to heterogeneity sensor nodes of energy levels. Our proposed method is a novel energy-efficient routing protocol in HWSN based on hierarchical topologies (EERHSN). Most of the heterogeneous network's algorithm considers two-level heterogeneous sensor nodes. Two-level heterogeneity in sensor node just covers a few sensor networks. Internet of things (IoT) is becoming popular nowadays. In IoT applications, there are different sensor node types (Kiani & Seyyedabbasi, 2018). The things in IoT can be sensor nodes, so different types of sensor nodes are used. As mentioned in the introduction section of the thesis (1.2) consider a smart home. In a smart home, there are many types of sensor nodes. The heterogeneity of this project affords us to propose a new heterogeneous wireless sensor network. This algorithm designed for three-level heterogeneity of the networks; in addition, EERHSN can be implemented for multi-level heterogeneity. Furthermore, the proposed method is a better protocol for real-time communication is wireless sensor networks. The clustering method is important to achieve energy efficiency. In this method, we consider a static clustering method. The clusters are not re-established in each round. As there are three types of sensor nodes in EERHSN normal, advanced and super nodes. Only advanced and super nodes transmit collected data to the base station/sink. As means, the advanced and super node act as cluster heads in EERHSN. These two types of sensor nodes have more energy than normal sensor nodes. Also, the super nodes have more than energy from advanced nodes. Since many routing protocols designed for wireless sensor networks. EERHSN is energy efficient and stable protocol for WSN.

This section presents EERHSN in a static clustering energy-efficient routing method with heterogeneous sensor nodes. In our network, we have three different types of sensors. Normal node, advanced node, super node, and these types of the sensor have heterogeneity in energy level, different transmission range. The EERHSN has two-phase. Clustering and routing phases.

3.2.2.1. Clusters and cluster heads

Figure 3.15 presents our network that has a total of sixteen regions in 100×100 (meters). In the clustering phase, as mentioned before we chose static clusters. There are sixteen regions that there are two types of cluster heads based on sensor nodes types: Cluster head and Region head. Advanced node and super nodes act as cluster head and region heads respectively. The cluster head is responsible for collect data packets from normal nodes. Region head should collect data from cluster heads. So, cluster and region heads in this approach are static and per round doesn't change. In the first phase, the whole of the area divided into four regions (Figure 3.15a). Then each region divided again into four sections too (Figure 3.15b, 3.15c). So totally, network has sixteen regions (Figure 3.15d).

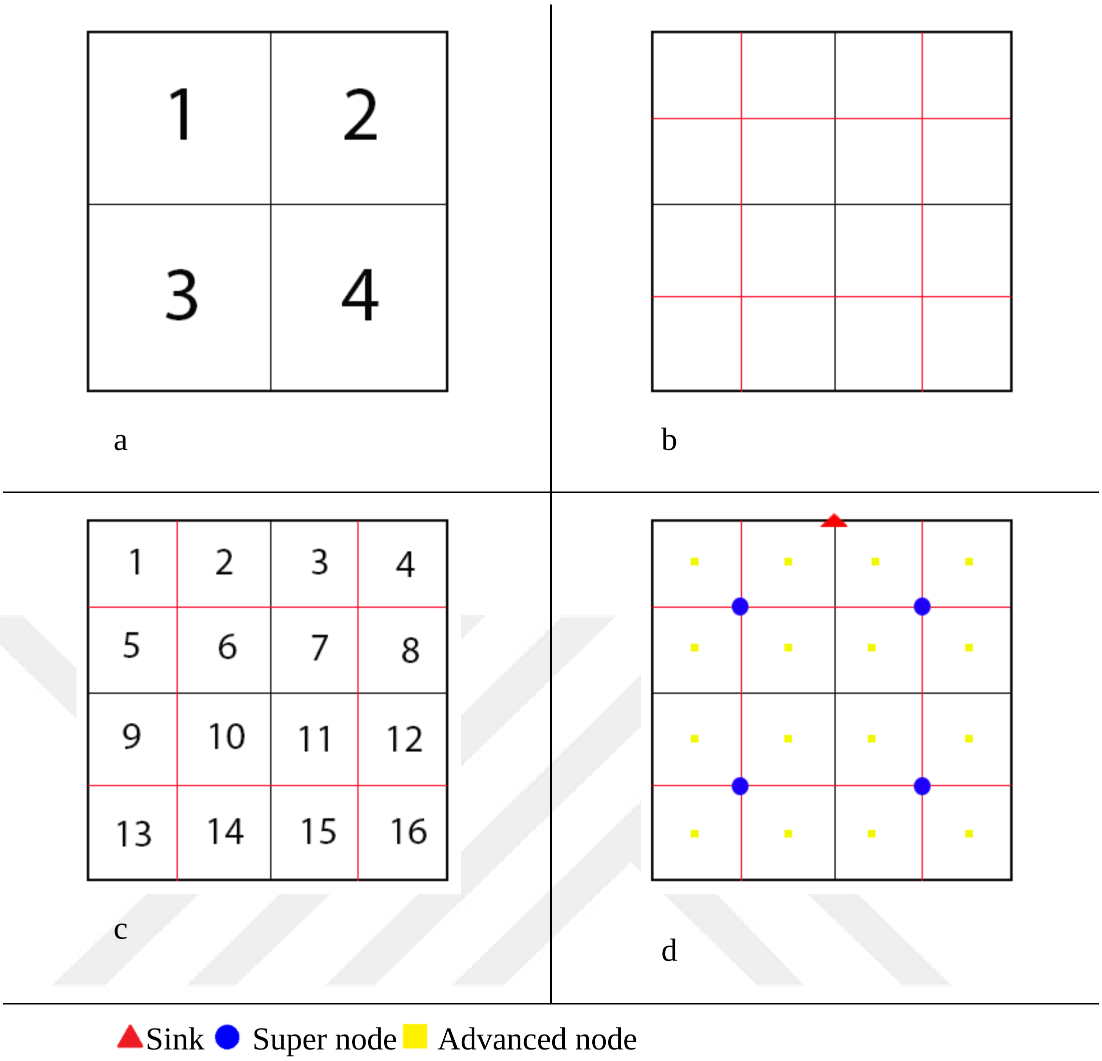


Figure 3. 15 EEHRSN network structure

In this phase, the location of advanced and super nodes is static. In the deployment step of the network, these two types of sensors located in determined locations. Normal nodes are deployed randomly (Figure 3.16).

Each of the sixteen regions has the same area, in the center of each region one advanced node located. Super node located in the first four divided regions according to Figure 3.15a. So totally, after deployment normal nodes, our network has four super nodes, sixteen advanced nodes. In other means, there are four region clusters, sixteen cluster heads.

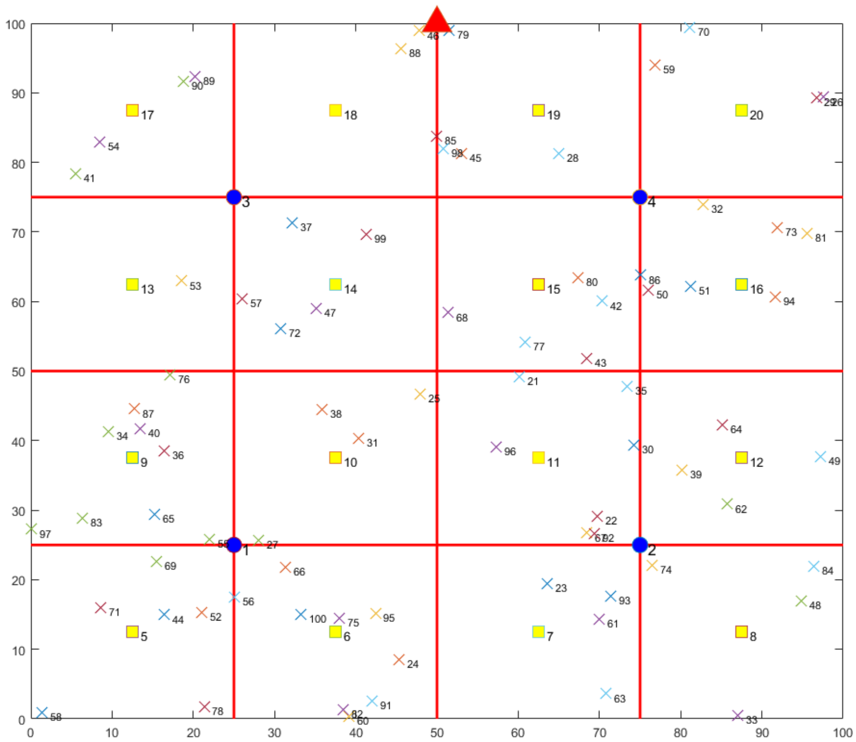


Figure 3. 16 Nodes deployment in EEHRSN

Here, we describe the features of each of the sensor nodes. Sensor nodes in EEHRSN have a difference in transmission range and battery level. The network has n number sensor nodes, that 4 number of sensor nodes are super node and sixteen of sensor nodes are the advanced node. We consider α, β parameters that α is a fraction of super nodes energy level and β is a fraction of advanced nodes. Super nodes battery level will be computed with this equation $E_0 \times (1 + \alpha)$ where E_0 indicates the initial energy of normal nodes. Also, the Equation $E_0 \times (1 + \beta)$is an advanced node energy level. Also Amount of energy consumption free space, amount of energy consumption for multipath fading, data aggregation energy, energy consumption per bit in the transmitter or receiver circuitry will be calculated like the use of α, β value.

3.2.2.2. Routing Algorithm

In EEHRSN, cluster and region heads located and normal sensor nodes are deployed. The next step is routing protocol in EEHRSN. The exact position of cluster and region heads is necessary to normal nodes in the routing phase. Our assumptions about different sensor node type are as follows:

- No mobility in nodes

- All nodes have a table that contains Euclidean distance between each other and sink/BS
- In all normal nodes, the neighboring table contains cluster IDs
- The normal node only can have a transmission with advanced nodes
- Coverage region of each advanced and super nodes declared
- All nodes have a unique identification number
- The base station is located at a fixed point and location can differ in any scenario
- The initial energy is fixed in each type of node
- Sink/BS is located in static position

In the initialization step, the cluster heads broadcast a message to appropriate sensor nodes (cluster members) that contain cluster head ID. Also, the super nodes (region heads) broadcast own ID to the advanced nodes. This information keeps on each sensor node table. The data collection scheme in EEHRSN is event-based. It means when a sensor node detects an event, it should be transferred event data to the related cluster head. The EEHRSN improves stability in the network. When a normal node dead or unavailable the advanced and super nodes can cover the network. So, connectivity in EEHRSN is at a high level. All the sensor nodes in this network equipped with unchangeable batteries. The sensor node should be consuming energy efficiently. Therefore, EEHRSN collects data in the event base.

In EEHRSN, there are some basic rules in data forwarding. Normal nodes cannot forward data packets to sink/BS directly. Advanced and super nodes act as a coordinator to forward data packets. Advanced and super nodes can track events when there are not alive normal nodes. All sensors except super nodes follow this rule. **First, tries to forward the data packet to appropriate cluster heads or region heads. Second, forwards the data packet to an upper level.** Super nodes just forward data packets to the sink/BS or other super nodes. It is important to note that, in transmitting data packets to other nodes, the transmitter controls the receiver energy level. Generally, the EEHRSN follows three steps of routing due to different sensor node types.

3.2.2.2.1. Normal nodes transfer data packets

In the first step normal node i ($21 < i < 100$) observed an event so it transfers data packet p on to the cluster head. As mentioned before, normal nodes just transfer data, packets to advanced nodes (cluster heads). Based on our assumption, normal nodes know the related cluster head ID. Therefore, the node i sends an advertising message to CH (Figure 3.17), to get information about cluster head energy level and traffic.

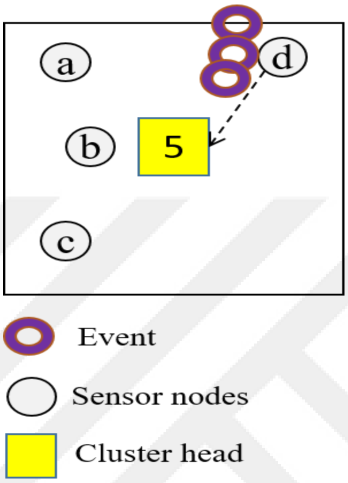


Figure 3. 17 Event detection in EEHRSN

Node ID	Cluster ID
a	5
b	5
c	5
d	5

Node ID	Request

Cluster ID	Energy	Traffic	Node ID

Node ID	Data	Cluster ID

Figure 3. 18 EEHRSN normal nodes routing

The cluster head reply node i message as (Figure 3.18). The node i checks traffic and energy level of the CH, if the CH has enough energy and is not busy, the node i forward data packet p to appropriate CH. For instance, according to Figure 3.18, node d knows the related cluster-ID is 5. First, try to forward data packet p to sensors node 5 based on a neighboring table. There is some exception here to control the related cluster head. If the related cluster head is unavailable or dead. Node i, looking for other cluster heads. Cluster head will be chosen from advanced nodes j ($5 < j <$

20). In this way, node *i* check, the acceptable distance between normal nodes and advanced nodes. If node *i* can forward data packet *p* to other cluster heads. On the other hand, if node *i* cannot find suitable CH based on acceptance distance, node *i* looking for up level advanced nodes. For instance, node *d* can forward the data packet to cluster-id *9* (Figure 3.18). This step, pseudo-code is in Figure 3.19.

Algorithm 1, Normal node data packet transfer

// distance: distance between Normal Node i and Advanced Node j
// dn: acceptable transmission distance between Normal Node and Advanced node
// E: Energy level of each node
IF ((distance<= dn) && (i.E >0) && (node i and node j is in the same region))
Packet p send to node j // node j is Advanced node
Else
Find in an upper level region of current advanced node and check
If ((distance<= dn) && (i.E >0))
Packet p send to node j // node j is Advanced node
End
End

Figure 3. 19 Overview of pseudo code for First Step

3.2.2.2.2. Advanced nodes transfer data packets

Cluster heads (Advanced nodes) collects data packets from related cluster members. Cluster heads should collect data from members then aggregate collected data. The aggregate data should be transfer to region heads (Super nodes). Here, the CH can choose two multi-hop paths with the exception. The main option is to forward the data packet to the appropriate region head. This path is the main path to data packet transmission. If the appropriate region head is not available, the other alternative path will be chosen. The first alternative way is to benefits other cluster heads. First, a cluster head checks the distance between own and sink/BS. If it is lower than the distance between the region head and sink/BS, forward the data packet to the sink/BS directly. In addition, if the region head is not available the cluster can do this. The second alternative path is benefiting cluster heads and region heads. In this scheme, the cluster heads forward data packet the up level or neighbors cluster head. Then that region super node (Region head) forward data packet to the sink/BS. In all of these options, cluster heads check the cluster and region heads energy levels and traffic before forwarding data. Figure 3.20 presents these alternative paths from cluster heads to the sink/BS.

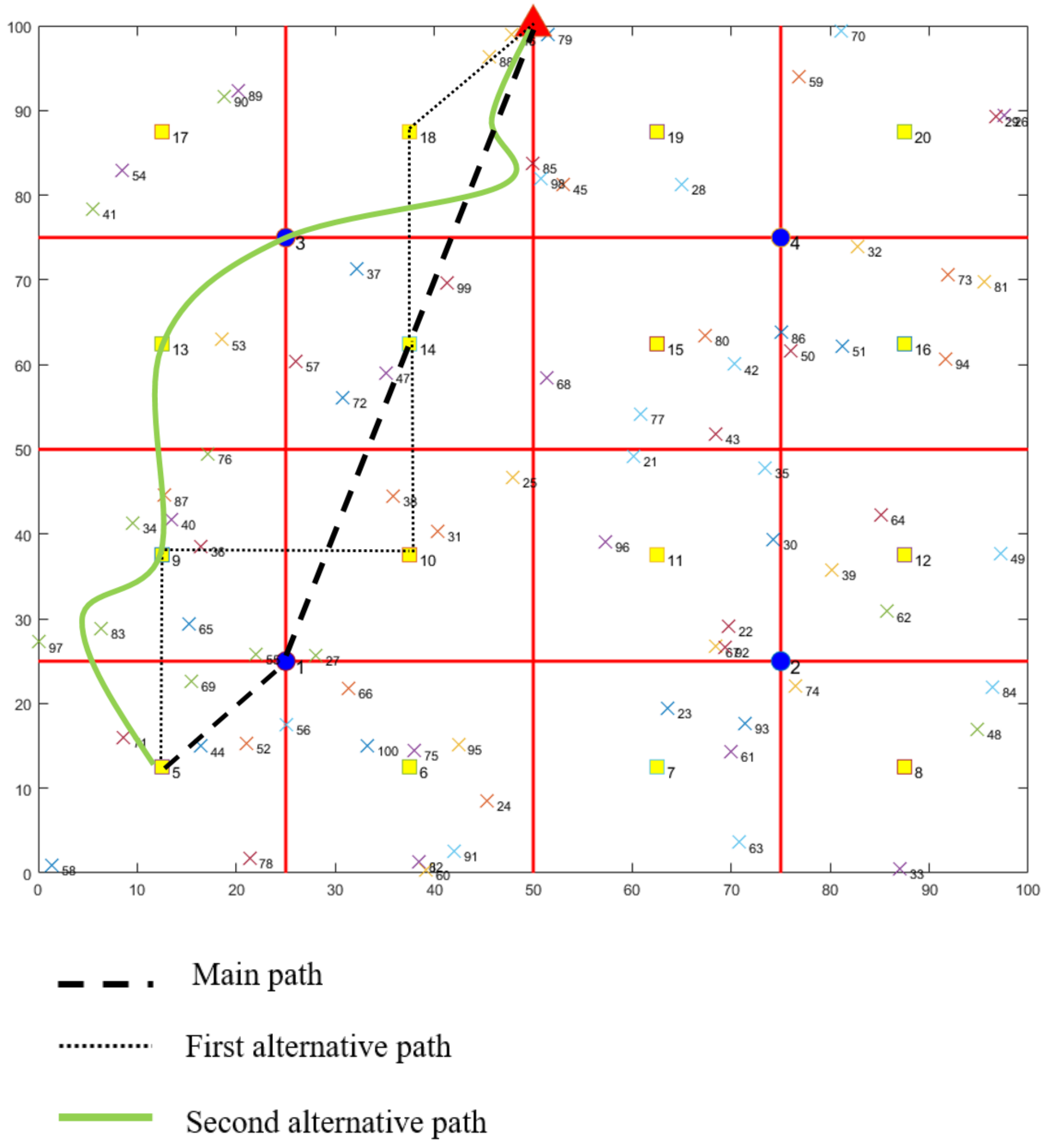


Figure 3. 20 Advanced node routing path in EEHRSN

The main and alternative paths for a transfer data packet from cluster head to the sink/BS can show as below:

1- CH → RH → BS

2- CH → CH → CH → CH → BS

3- CH → CH → CH → RH → BS

For each alternative path, the cost function first computes the cost. Then, the path with the lowest cost will choose the best and optimal path to forward the data packet. The following Equation computes the cost of CH to BS (Equation 3.5).

$$Cost_{CHtoBS} = \sum_{i=1,i\in n}^{BS} distance_{i,BS} + \sum_{j=1,j\in n}^{BS} Energy_{j,BS} + \sum_{k=1,k\in n}^{BS} Traffic_k \quad (3.5)$$

According to the Equation 3.5, the Euclidean distance between each of the sensor nodes (cluster and region heads) is computed. Also, the power dissipating from each sensor node (cluster and region heads) in data transferring obtained from Equation 3.6. In addition, the traffic of receiver and the sink/BS obtained from the broadcasted message. The power dissipated while sending k bits of data can be calculated as:

$$E_{tx}(k,d) = \begin{cases} E_{elec} \times k + e_{fs} \times k \times d^2 & ,d<d_0 \\ E_{elec} \times k + e_{mp} \times k \times d^4 & ,d\geq d_0 \end{cases} \quad (3.6)$$

And to receive *k* bit of data radio expends:

$$E_{rx}(k) = k \times E_{elec} \quad (3.7)$$

Where:

- E_{elec} is the amount of energy consumption in the transmitter or receiver
- e_{mp} is multipath fading energy consumption
- e_{fs} is free space energy consumption and d is the distance between the nodes

After all, the path with minimum cost is selected to forward data packet p with k bit to sink/BS.

Cluster ID	Region ID
5	1
6	1
7	2
8	2
…	…

Figure 3. 21 Example of cluster information table EEHRSN

As mentioned before advanced node (cluster head) j $5 < j < 20$ can transfer data packet p in main and alternative paths to the sink/BS. Here, there is an example. In the main path. Cluster head j broadcast a message to request the energy level and traffic situation of the region head $k, 1 < k < 4$ (Figure3.22). After this, the region cluster replies to the cluster head request as Figure3.22.a. Then, cluster head j forward data packet to the region head. In this way, EEHRSN does not compute the cost because this path is the main path. So the data transmission operated from this path. In exception situations like region head traffic is high or drained energy supply. Two alternative paths chose from Equation 3.5 to compute the path's cost.

a)

Cluster ID	Request

b)

Region ID	Energy	Traffic	Cluster ID

c)

Cluster ID	Data	Region ID

Figure 3. 22 Advanced node data packets in EEHRSN

All the cluster head keeps the appropriate region head identify numbers. In two alternative paths also in the main path, this information from the region head helps the cluster head (Figure 3.20). EEHRSN also improves reliability or fault tolerance. Reliability improves packet delivery in the network; also reduce the lost packets number. On the other hand, this is leach to enhance network connectivity. It means when normal nodes dead in a cluster, the advanced nodes can cover the cluster as much as possible. Also, the alternative paths in routing data packet from cluster heads to sink/BS cause to find new routing paths. Figure 3.23 presents the pseudocode of the advanced node data packet transfer.

```
Algorithm 2, Advanced node data packet transfer

//k.E is the energy level of the region head that cluster head  requested
// Traffic is the traffic situation of the region head that cluster head  requested
//there are three path to forward data from CH to BS. The first one main path two
  other alternatives
If (K.E>0) &&(Traffic is available)
Packet P send to region head k
Else
Compute the cost between two alternative paths; Equation 3.5
//First alternative path is: forward data packet from cluster head to sink/BS
//Second alternative path is: forward data packet from cluster heads and other region
  head to sink/BS
Find minimum cost(Cost_CHtoBS )
Packet P send to sink/BS
End
```

Figure 3. 23 Overview of pseudo code for Second Step

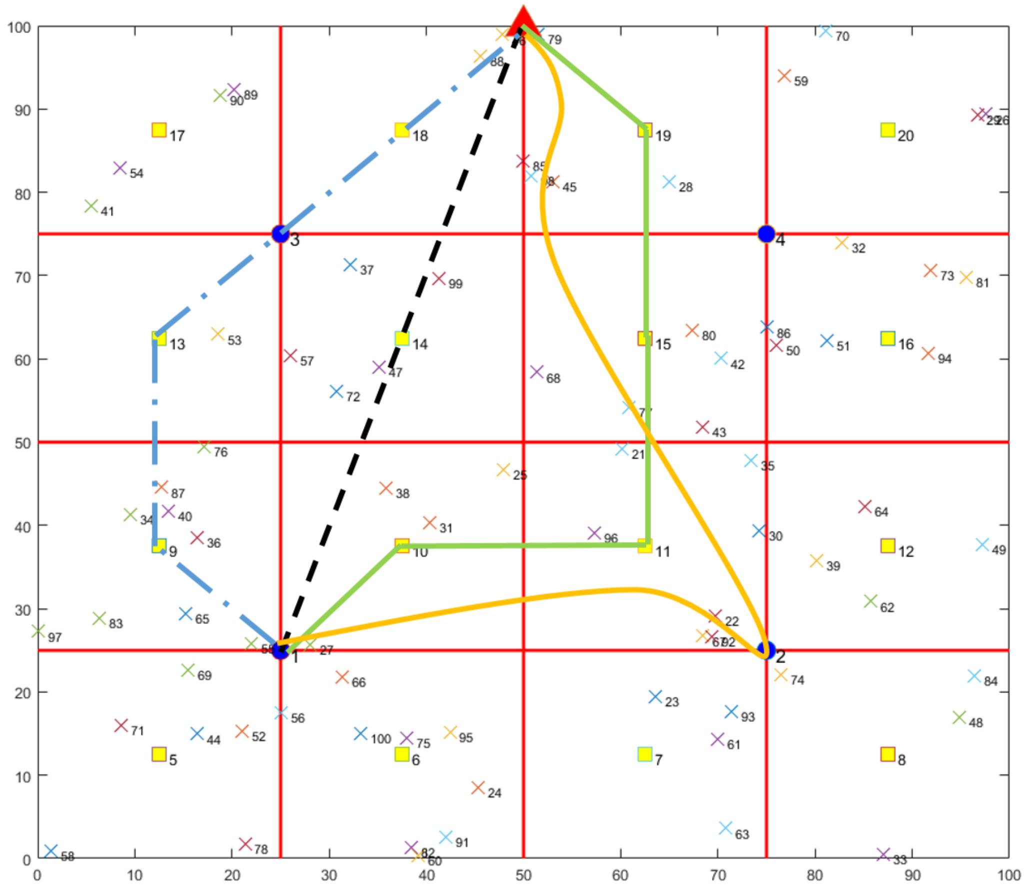
3.2.2.2.3. Super nodes transfer data packets

As mentioned before, the super nodes (region heads) are responsible for collect data from advanced nodes (cluster heads). Therefore, they should have more energy than other sensor nodes. In assumptions of EEHRSN, the super nodes have a time more energy. Also, the communication range of super nodes is higher than in others. In the proposed network of EEHRSN, there are only four super nodes. Each of this located in a specific position to cover the region's cluster heads. After all, the super nodes deliver most of the data packets to the sink/BS. While the advanced nodes can deliver data packets to the sink/BS individually. **The main path** is using super node to transmit the data packet directly to the sink/BS. Also in **the first alternative path** transfers data packet to the appropriate neighbor region head, then that super node continues transmission. Assuming that, when normal and advanced nodes dead, the super nodes can cover the region as much as possible to detect events. This is the strength of EEHRSN that advanced and super nodes can detect events in the last iterations of the networks. As said before, the reliability and fault tolerance in EEHRSN is one of the main advantages. There **are other two alternative paths** for super nodes to transfer data packets to sink/BS. The main path is to transfer in a single hop. Since the super nodes benefit high transmission range. They can easily send data packets to the sink/BS. Alternative paths try to increase the data delivery rate. When the super node does not have enough energy to send data packets at a high transmission rate, it benefits suitable advanced and super nodes. The **second alternative path** is to transfer data via advanced and super nodes. In this way, super node sends the data packet to up-level advanced nodes. The advanced node continues via advance nodes until delivering the data packet to the sink/BS. In **the third alternative path,** the advanced node transfers the data packet to the up-level super node. Then it transfers data packet in a single hop to sink/BS.

Like advanced node’s main and alternative paths, the Equation 3.8 computes the cost of each alternative path. After that, the EEHRSN decides to choose which path. As mentioned before, EEHRSN increases data reliability also tries to reduce energy consumption. As in a heterogeneous wireless sensors network, there are different energy level sensor nodes, the protocols should be energy efficient. Super nodes like advanced nodes try to control node's failure. This approach with one main path and three alternative paths can decrease failures in the network. As a result, the advanced and super nodes in the absence of the normal nodes have the sensing functionality as much as possible to cover regions. The following equation calculates the cost of alternative paths between RH and sink/BS.

$$Cost_{RHtoBS} = \sum_{i=1,i\in n}^{BS} distance_{i,BS} + \sum_{j=1,j\in n}^{BS} Energy_{j,BS} + \sum_{k=1,k\in n}^{BS} Traffic_k \quad (3.8)$$

Where the Euclidean distance between the cluster head and region heads along the path. Also, the energy of the sensor nodes along in the path has calculated the Equations 3.7, 3.8. The sink/BS and each sensor node along the path traffic control are executed. After this, the minimum cost path is selected to the alternative path between region head and sink/BS.



Main path

First alternative path

Second alternative path

Third alternative path

Figure 3. 24 Super node routing path in EEHRSN

The main and alternative path follow these routing paths:

1- RH → BS

2- RH → RH → BS

3- RH → CH → CH → RH → BS

4- RH → CH → CH → CH → CH → RH → BS

Region ID	Request

Region ID	Energy	Traffic	Region ID

Region ID	Data	Region ID

Region ID	Energy	Traffic	Cluster ID

Cluster ID	Data	Region ID

Figure 3. 25 Super node data packets in EEHRSN

In the last step node $k, 1 < k < 4$ want to send packet p to sink/BS. There are four paths in routing the data packet p. The main path is to send data packets directly to the sink/BS. The first alternative path is that node k, benefits the remaining k super nodes. The second alternative path uses advanced nodes j, $5 < j < 20$ and super nodes. For instance node, k first sends data packet p to node j. node j sends the data packet to up-level advanced nodes j then transfer to a super node. Finally, super node transfer data packet to the sink/BS. In the third alternative path benefits just advanced nodes ($5 < j < 20$). Initially, super node transfer packet p to the advanced nodes. The advanced nodes decide to continue via the same type of sensor nodes. At the end deliver data packet p to the sink/BS (Figure 3.25). The cost function in the alternative paths helps EEHRSN to decide the path. In the following, the pseudo-code of the super node's transmission is described (Figure 3.26).

Algorithm 3, Super node data packet transfer

//k.E is the energy level of the region head that region head requested
// Traffic is the traffic situation of the sink/BS that region head requested
//there are four path to forward data from RH to BS. The first one main path three other alternatives
//E: Energy level of each node
If (K.E>0) && (Traffic is available)
Packet P send to region head k
Else
Compute the cost between two alternative paths; Equation 3.8
//First alternative path is: forward data packet from neighbor region head to sink/BS
//Second alternative path is: forward data packet from cluster heads and other region head to sink/BS
//Third alternative path is: forward data packet from cluster heads to sink/BS
Find minimum cost ($Cost_{RHtoBS}$)
Packet P send to sink/BS
End

Figure 3. 26 Overview of pseudo code for third Step

3.2.3. Simulation

EEHRSN method is simulated also in MATLAB, also the all algorithms in this chapter are simulated in the same situation in MATLAB. In the simulation, we obtained the output metric in the network such as; network lifetime, packet delivery ratio and network balance. The algorithms that compared with EEHRSN are SEP (Smaragdakis et. al., 2004), DEEC (Qing et. al., 2006), EECH (Kumar et. al., 2009). Also, the input parameters are the same as each algorithm. The input parameters are described in table 3.7.

Table 3. 7 configuration parameters for EHHRSN

Parameters	Values
Network Size	100*100 m
Normal node(N1)	80
Advanced node(N2)	16
Super nodes(N3)	4
Base station location	50*100 m
Data packet size	4000 bits
e_0	0.5 J
e_{fs}	10 pj/bit/m2
e_{mp}	0.0013 pj/bit/m4
e_{elec} (TX,RX)	50 n j/bit
Data Aggregation Energy cost	50 n j/bit

As there are three types of sensor nodes in the EEHRSN. They are different energy levels. As mentioned before, there are two control parameters α, β. Respectively, they determine advanced and super nodes energy fraction. We consider $E_0 \times (1+\alpha)$ and $E_0 \times (1+\beta)$ to advanced and super nodes initial energy. Total we consider *n* nodes in the network. In the simulation $\alpha = 1, \beta = 2$. Consequently, the total energy of the network is:

$$E_{Total} = (N1 \times E_0) + (N2 \times (E_0 \times (1+\alpha))) + (N3 \times (E_0 \times (1+\beta))) \quad (3.9)$$

This equation determined that the energy level in the EEHRSN has heterogeneity and energy is distributed respect to responsibilities of each node. As mentioned before, initialization of the network with determining the cluster and region heads position. Normal nodes are deployed randomly. This network configuration and assumption are the same for all comparison algorithms. The sink/BS is located in 50 ×

100 meters. Also, all of the sensor nodes are heterogeneous and data packet size per round is 4000 Kbits. The coverage range (cr_n) for normal sensor nodes is 10 meters. Advanced and super nodes coverage range follows the α, β parameters.

$$cr_a = cr_n \times \alpha \text{ , } cr_s = cr_n \times 2\beta \quad (3.10)$$

As mentioned before, the EEHRSN is an event-based network. We execute the network in 500 rounds that each round takes 2 seconds. Totally, the network takes 1000 seconds. Besides, 200 events are randomly created each event takes about 5 seconds. The range of each event is 4 meters.

We evaluate both algorithms in terms of performance that mentioned previously. Also, all of the sensor nodes are static. All nodes follow the specific architecture based on own type. As the main features of the sensor nodes, the energy level is limited except for the sink/BS. The EEHRSN protocol consisted of two steps, clustering, and routing. The clustering method is done in network initialization. All normal sensor nodes know the appropriate cluster head ID. Also, the cluster heads know the appropriate region ID. The routing step has three different steps based on the nodes types. The main goal of the EEHRSN to reduce energy consumption in heterogeneous wireless sensor networks and node failure in data transmission.

3.2.4. Experimental Results

This section evaluates the performance of EEHRSN in simulation. We same algorithms in the same configuration parameters. The evaluation metric parameters are network lifetime, packet delivery ratio and network balance. The comparison algorithm such as SEP, DEEC, and EECH also simulated in MATLAB. Also, the input parameters as described in Table 3.7 is the same.

Figure 3.27 shows the number of alive nodes from 100 sensor nodes in the along of network duration. As shown in Figure 3.27, the EEHRSN has a few alive nodes after the network end. It means this network able to continue after 1000 seconds too. Since there are different types of sensor nodes with different energy levels. As mentioned in the algorithm description, after normal nodes dead other types of sensors able to detect events. Generally, in event-based networks, energy consumes at specific times. Like TEEN (Manjeshwar et. al., 2001a) algorithm, the EEHRSN is event-based. In Figure 3.28 we compare the performance of TEEN and EEHRSN algorithm, but they have a difference in sensor types. TEEN is a homogeneous wireless sensor network while EEHRSN in a heterogeneous wireless sensor network.

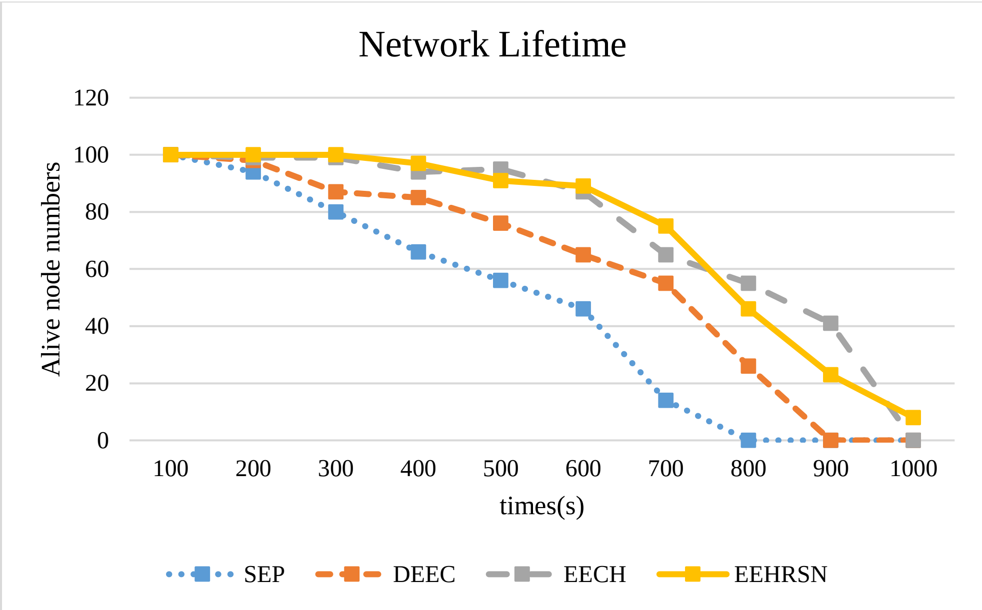


Figure 3. 27 Number of alive nodes in EEHRSN

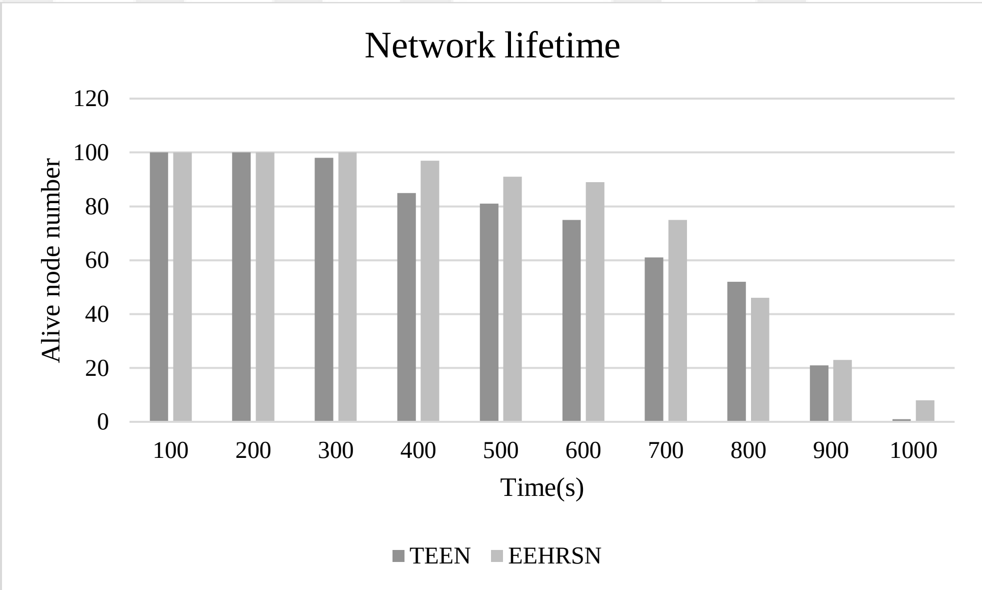


Figure 3. 28 Comparison of TEEN and EEHRSN algorithms

The number of alive nodes in the TEEN algorithm is better than in comparison to other heterogeneous wireless sensor networks like SEP, DEEC, and EECH. While the sensor node is not the same. EEHRSN also keeps alive nodes until the last of the network. It is necessary to mention that, the input configuration parameters for TEEN and EEHRSN are the same. This comparison was performed because both algorithms are event-based.

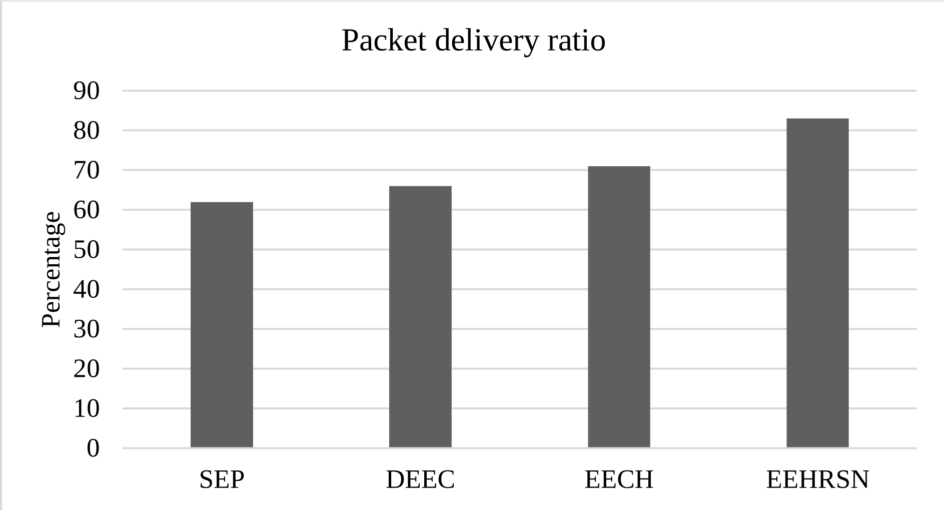


Figure 3. 29 Comparison of packet delivery ratio

Figure 3.29 shows the percentage of successful data packets received by the sink/BS. Packet delivery in the EEHRSN is better than other algorithms. EEHRSN was able to receive 81% of the data packets. The amount of difference EEHRSN with SEP, DEEC, and EECH are 21, 17 and 12 percent respectively.

The clustering method is the first step of the EEHRSN method. Clusters are static in the EEHRSN, so the cluster members do not consume energy in extra communication to explore the cluster head. In this, we can reduce further overhead in the network. Most of the clustering methods drain sensor nodes in for cluster head selection. In each round, all sensors should take part in the cluster head selection step. As the clustering method is a good way to achieve energy efficiency in the WSN. However, we consider with a heterogeneous sensor that benefits more energy, it is not necessary to change cluster head in each iteration. As shown in Figure 3.30 the cluster in the EEHRSN is static and other algorithm clusters are changed in each iteration based on the cluster formation scheme.

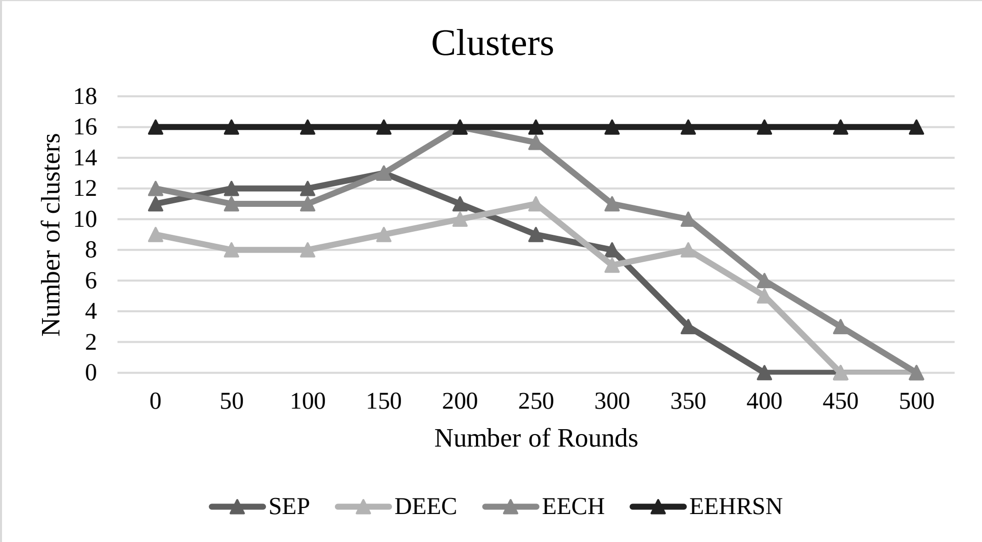


Figure 3. 30 Number of cluster heads in each algorithm

3.2.5. Summary and Conclusion

EEHRSN algorithm is a novel energy-efficient method in heterogeneous wireless sensor networks; also it is an event-based protocol. Three are three different sensor types in the network. The protocol has two steps, clustering, and routing. The clustering scheme is a few different from most of the clustering algorithms like LEACH and SEP. The routing step based on three heterogeneous sensor nodes. The sensor nodes have different energy levels, communication range, and computation memory. The sensor nodes in this network know their position. The main goal of the EEHRSN is to prolong the network lifetime.

The clustering scheme in this protocol follows a static way. In the initialization of the algorithm, the network divided into four equal regions. Then each of these regions divides into four regions again. The network has sixteen virtual regions. In the EEHRSN, there are three types of sensor nodes named; normal nodes, advanced nodes, and super nodes. Respectively, these nodes have different energy levels and communication ranges. Normal nodes are responsible for collect and cover appropriate clusters. Advanced nodes act as cluster heads, collect and aggregate the data packets from normal nodes. Super nodes are region heads in the network that collect aggregated data from advanced nodes (cluster heads) then transfer to sink/BS. There are eighty normal nodes, sixteen advanced nodes (cluster heads) and four super nodes (Region heads). The super nodes (region heads) are located in the center of the first divided regions. Advanced nodes (cluster heads) is located in the center of sixteen regions. In other means, there are twenty cluster and region heads. These node's energy level is different. Advanced and super nodes are more than normal nodes infraction of two parameters α, β respectively.

In the routing step, the EEHRSN proposed an energy-efficient approach to reduce overhead. Also, EEHRSN increases data packet delivery. The routing protocol for each sensor type varies from others. The normal nodes just transfer data packets to advance nodes (cluster heads). In the initialization of the network, the normal nodes know the appropriate cluster head ID. Also, if the appropriate cluster head is not available the normal nodes can transfer data packet to neighbor cluster head of course with some consideration. The cluster heads should transfer the aggregated data packet to the appropriate region head. Also, they know the region's head ID. As EEHRSN tries to improve data packet delivery and balance fault tolerance. EEHRSN considers two alternative paths to transfer data packets to region heads and sink/BS. The advanced nodes can check the region heads availability. In the first alternative path, if there are not suitable region heads, the advanced node transfers data packet through other advanced nodes (cluster heads). EEHRSN considers that advanced nodes deliver the data packet to the sink/BS. The second alternative path is benefiting cluster heads and other region head. Transfer data packet to neighbor cluster head, then it sends the data packet to sink/BS via an appropriate super node (region head). Also, the super nodes have a specific routing protocol. In the EEHRSN, the packet delivery to sink/BS is responsible for super nodes, but it can be changed in unexpected situations. The main path to transfer data packet from region head to the sink/BS is to send data packet in a single hop. If it has not enough energy to transfer in high range communication, the region head should follow three alternative paths. Transfer via other region heads, transfer only by advanced nodes or benefits advanced nodes and a region head. It must be mentioned that there are cost functions for both cluster and region heads to choose the minimum cost path.

The simulation and result comparison section widely are discussed about the obtained results. All of the input configuration parameters are the same for all simulated algorithms. The simulation is executed in the same network area. The observation proved that the EEHRSN is increased the network lifetime. While there are, some alive nodes after the 500 iterations are finished in the network simulation. In the EEHRSN after normal nodes dead, the advanced and super nodes can cover the area as much as possible to detect events. There is a homogenous wireless sensor network algorithm that is event-based also (TEEN). This algorithm is compared with EEHRSN. The network lifetime of both of them is better than others, while the EEHRSN again has good performance. The number of alive nodes in EEHRSN is higher than a TEEN algorithm.

In data packet delivery, the EEHRSN has noticeable performance. Since it has a long network lifetime. Also, the sensor nodes have the same functionality in event detection. As mentioned before, after normal nodes dead, the advanced and super node can detect events in own coverage area. This scheme is performed after all normal nodes of death. If the advanced and super nodes same as normal nodes detect an event, consequently they energy is drained easily. The goal of the EEHRSN is to increase network life so this mechanism performed logically. On the other hand, when there are alive nodes in the network and they performed own functionality, the data packet delivery ratio is increased. The simulation results clear that 81% of the data packets from sensor nodes is received by sink/BS.

In the clustering schemes, the number of clusters is important. EEHRSN follows the static clusters so the cluster numbers are stable from initialization until the last, but the cluster heads number has a difference from the first iteration until the end. Also, the EEHRSN has good performance in the number of cluster heads, because the cluster heads are in this algorithm and have more energy than normal nodes. This caused to cluster heads is more stable even in the 95% of last iterations.

After all, the EEHRSN has good performance in a network lifetime. This algorithm has heterogeneity in the sensor nodes. It is a good protocol to apply in IoT application routings. Most of IoT applications have limited sensor nodes. It means the scale of IoT application is small-scaled but has different sensor types. EEHRSN can have good performance in IoT projects since it has good performance in small-scale networks. Besides, EEHRSN is a good choice for real-time applications. EEHRSN has few packet delays; also, the reliability of this protocol is better than others. In real-time applications, reliability is an important factor.

3.3. Improved version of the HEEL algorithm (I-HEEL)

3.3.1. Introduction and Motivation

The wireless sensor networks have consisted of many sensor nodes and one or more than one sink/BS. In most WSN the sensor nodes resources are limited as power and communication range, the sink/BS is benefited the unlimited energy supply. There is plenty of research to solve these challenges in sensor nodes. As sensor nodes have progressed in technology like power, range, material but still the power limitation unsolved. So the researchers focused on the application layer of WSN to improve energy consumption by protocols such as routing algorithms, data collection methods, duty cycle techniques, topology controls, and clustering methods. Also, the network topology is important in energy efficiency methods. As mentioned before, the network topology is categorized into three; flat-based, hierarchical-based and location-based networks.

This chapter describes the clustering method. Actually, this method is an improved version of the HEEL algorithm explained in section 3.1. As mentioned before, the cluster head selection is the main challenge in the clustering scheme. In the HEEL algorithm cluster head selection performed by four parameters. Such as; residual energy, the distance between a sensor node and the base station and the number of links to neighbors. These parameters have effects for cluster head selection with coefficients between (0, 1). The fitness function selects the optimal cluster head with these parameters.

In HEEL, the coefficient for each parameter is static and it is chosen by simulation experience. This is one disadvantage of the HEEL algorithm. I-HEEL algorithm is the improved version of the HEEL algorithm by metaheuristic algorithms. As mentioned in section 2.2.5 we proposed two metaheuristic algorithms. Generally, the proposed metaheuristic algorithms are modified version of the grey wolf optimizer (GWO) (Mirjalili et. al., 2014). I-GWO and Ex-GWO algorithms are our proposed methods. Metaheuristic algorithms are explained widely in section 2.2.

Metaheuristic algorithms initially have a search space to find optimal solutions. The population is important in algorithms. For instance, in the GWO algorithm, the number of wolves or population is defined in the initialization step. The search element or population search for the best solution based on the algorithm. Most of the metaheuristic algorithms inspired by nature or physics rules. In the GWO algorithm search of the best solution based on the prey position. The wolves categorized on alpha (α), beta (β), delta (δ), and omega (ω) wolves. There is a hierarchy in the pack of grey wolves. Each of them has responsibility on the pack. The first three wolves just looking for a hunt. Also, these wolves have functionality in encircling, hunting and attacking the prey. Each of these functionalities has a mathematical equation in the GWO algorithm. In the GWO all of the behaviors of the grey wolves in simulated.

I-HEEL is an improved version of the HEEL algorithm. I-HEEL has consisted of two steps. Updates coefficients and cluster head selection. In the update coefficients step, the metaheuristic algorithms try to find the best optimal coefficients. For instance, if the coefficients supposed like 0.45 for node energy, 0.15 for the energy of node's neighbors, 0.25 for the number of hops to the base station and 0.15 for the number of links to neighbors. In this case, there will be a steady rule along the network to select a cluster head. The proposed I-HEEL algorithm avoids this drawback by metaheuristic algorithms.

The main goal of the clustering strategy is to use energy resource efficiently. In clustering model, sensors are organized into clusters and logically it is a hierarchical strategy. At least one sensor node is used in the manufacturing of cluster. The CH is responsible for the cluster to collect and aggregate data from cluster members in dividing time slots (TDMA) mode. The aggregated data forwards from CHs in single-hop or multi-hop to the base station (BS). Generally, in the cluster method topologies, the aggregated data transferred via CHs to BS. Also, cluster members switch between sleep and awake modes depend on protocol. The designing routing algorithm is deepened on the application and network topology (Gautam & Pyun, 2010; Hammoudeh & Newman, 2015). Communication between CHs and cluster members is important in network lifetime. Sometimes, the closest CHs to BS consume more energy than others. Therefore, in the homogenous network, it is a critical issue. After all, the clustering strategy consists of two steps, cluster formation, and data transmission phase. In the cluster formation phase, CHs are selected and CMs get appropriate CH's identification number (Figure 3.31). Selecting suitable CHs is a key step to balance energy level and network lifetime. In the clustering strategy, after the cluster formation step, the data must forward from CHs to BS. In the data transmission step, CHs are responsible for forwarding collected data to BS, generally, CH forward the data to the BS via the upstream CHs to BS in single-hop or multi-hop way.

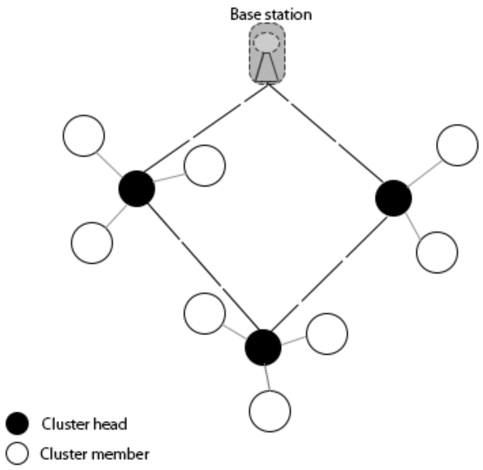


Figure 3. 31 Example of Clustering method

In the cluster head selection, as same as the HEEL algorithm, there is a fitness function. The main difference is in the I-HEEL the coefficients are dynamics along the network lifetime. In the cluster head selection, I-HEEL also follows four parameters to find the optimal CH. In the I-HEEL network topology, there are sixteen clusters. Each cluster has the sensor nodes as cluster members. They collect data from the region and transfer data to own CH. At the end of each iteration in the network, the metaheuristic algorithms choose the CH.

The hierarchical based protocols have better productivity in energy than other protocols. I-HEEL is also based on this type of protocol. As mentioned before, hierarchically based protocols consisted of cluster and cluster heads. There are different models to select cluster heads and divide the network into clusters. Also, most cluster head models consider residual energy, distance, and traffic in the decision step. The well-known clustering models are described in chapter 2.

In the I-HEEL, the main aim is to select optimal CH. This causes to have an enhancement in the network lifetime. The cluster heads are changed in each of the rounds. The cluster head selection increases the clustering overhead in the network. We consider static clusters to reduce overhead as much as possible. I-HEEL is reduced the number of dead nodes. As a result, the proposed method improves the network packet delivery ratio and connectivity. The routing of data packets in the I-HEEL is based on the TDMA schedule. The cluster members transfer data packets to the CH in the TDMA schedule. Then the cluster head receives data packets and aggregates these. The aggregated data send from CH to the sink/BS in TDMA schedule too. The I-HEEL method description is explained in the following section.

3.3.2. I-HEEL Algorithm Description

In this section, we describe the improved version of the HEEL algorithm. The HEEL algorithm is a novel method to select the cluster heads in each round based on four parameters. Residual energy, the distance between a sensor node and the base station and the number of links to neighbors are the parameters that have an effect in cluster head selection. In the HEEL method, the cluster head selection has four coefficients to choose the best cluster head. The coefficients are selected based on simulation experience and there are static in all of the network lifetimes. In the improved version of the HEEL algorithm, the coefficient chooses by metaheuristic algorithms. As in this thesis, the author proposed two improved versions of the GWO algorithm. I-GWO and Ex-GWO are two metaheuristic algorithms applied in the I-HEEL algorithm.

I-HEEL method is a cluster head selection method. I-HEEL network protocol based on hierarchical protocols. In hierarchically based protocols the clusters and cluster head is an important phase. Here in the I-HEEL, there are sixteen static clusters. Each cluster should have at least one node in the network initialization. The cluster head selection executes in each round, it has two options, changes the current cluster head or continues by the current cluster head. After cluster head selection, cluster members should be aware of the clusters' cluster head ID. The selected cluster head broadcast a message containing the cluster head ID to cluster members. The clustering scheme, manage resources in the network efficiently especially the battery level. The clustering scheme also reduces the transferring between sensor nodes and the sink/BS. The cluster head selected to reduce the communication and the CH is responsible for the data transferring to the sink/BS in single or multiple hops.

The clustering method performance is based on finding the best cluster head. In the proposed method, the author attempts to apply the proposed I-GWO and Ex-GWO algorithms in finding the best and optimal CH. I-HEEL consisting of two major steps. First, update the static coefficient dynamically in each iteration. Second is find the best and optimal cluster head from candidate sensor nodes. As mentioned before the fitness function in the HEEL algorithm has static coefficients. I-HEEL tries to choose the coefficient based on the current round situation of the network, and then chose the best cluster head.

3.3.2.1. Update Coefficients in I-HEEL by GWO, I-GWO and Ex-GWO

I-HEEL's main goal is to choose the coefficients of the HEEL algorithm by the metaheuristic algorithm. As in the previous section described metaheuristic algorithms like GWO, I-GWO and Ex-GWO find the best coefficients in each round by the network parameters. Node energy, the energy of node's neighbors, number of hops and number of links to neighbors are the network parameters. In the proposed algorithm, these parameters very important in choosing a cluster head in each cluster. So choosing static coefficients can be not suitable. As a result, the coefficients will be steady along with the network. Here, the I-HEEL updating coefficients are described widely.

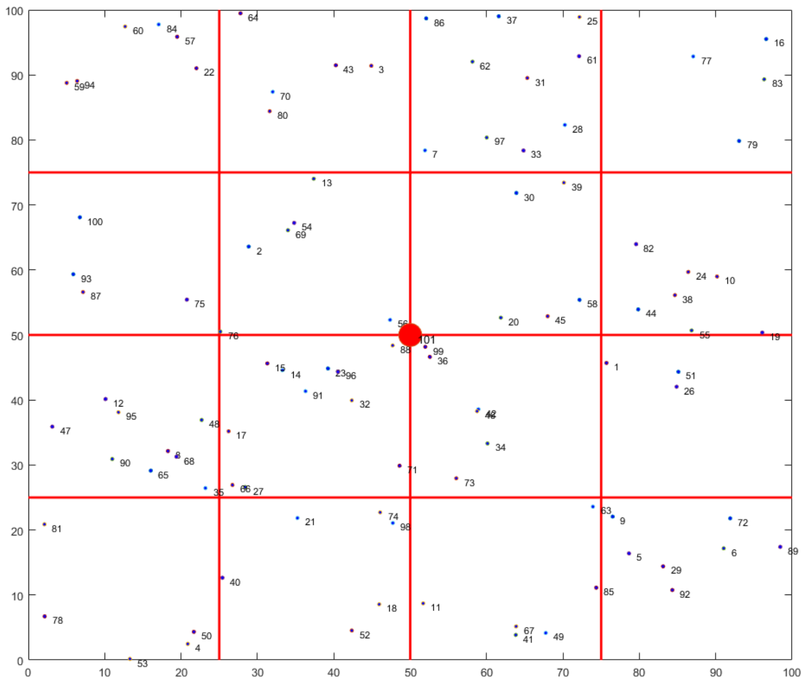
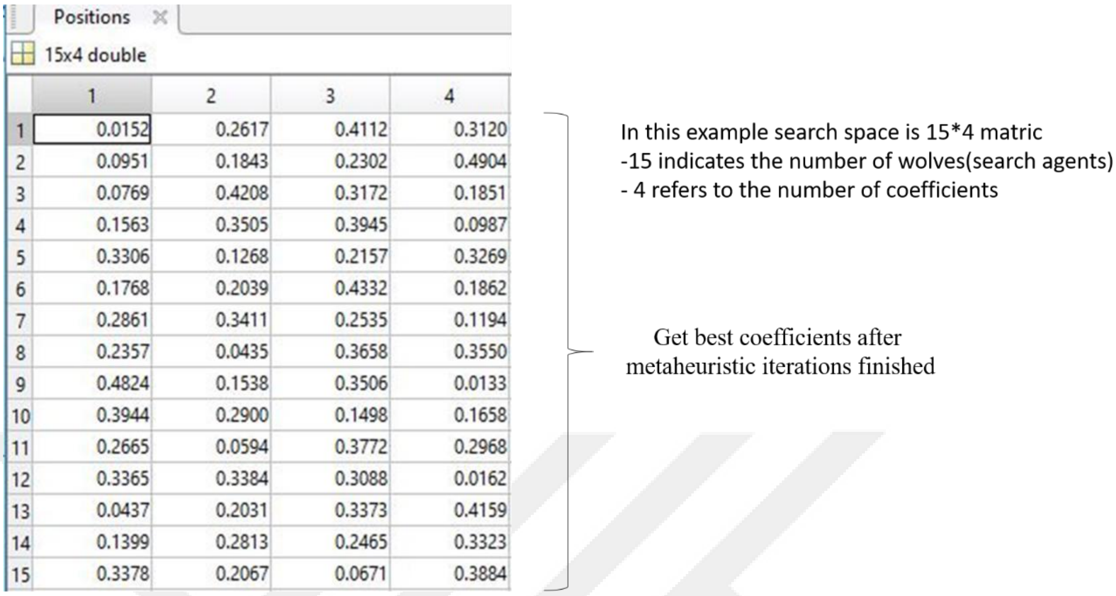


Figure 3. 32 The network structure of I-HEEL

In the proposed method, the search agent's numbers are 15. It means there are fifteen wolves in the algorithm that tries to find the best coefficients. As in the metaheuristic algorithms, there are random parameters; we declare the maximum iteration run in the algorithm. In the first round of the network, the metaheuristic algorithm declares a search space. The search space matrix has four columns and fifteen rows. Columns and rows based on the parameters and search agents respectively. For each of the coefficients a_1, a_2, a_3, a_4 a random number are assigned between 0 and 1, while $a_1 + a_2 + a_3 + a_4 = 1$. (Figure 3.33)



	1	2	3	4
1	0.0152	0.2617	0.4112	0.3120
2	0.0951	0.1843	0.2302	0.4904
3	0.0769	0.4208	0.3172	0.1851
4	0.1563	0.3505	0.3945	0.0987
5	0.3306	0.1268	0.2157	0.3269
6	0.1768	0.2039	0.4332	0.1862
7	0.2861	0.3411	0.2535	0.1194
8	0.2357	0.0435	0.3658	0.3550
9	0.4824	0.1538	0.3506	0.0133
10	0.3944	0.2900	0.1498	0.1658
11	0.2665	0.0594	0.3772	0.2968
12	0.3365	0.3384	0.3088	0.0162
13	0.0437	0.2031	0.3373	0.4159
14	0.1399	0.2813	0.2465	0.3323
15	0.3378	0.2067	0.0671	0.3884

Figure 3. 33 Search space of coefficients

This table will be updated in each round of the network. It must to be clear that the iteration of the metaheuristic algorithms is different the rounds of the network. In the I-HEEL, the iteration number for each solution is 10. The whole network rounds are 500. As mentioned before the network has fixed clusters. Each clusters should be have a cluster head. In the sensor deployment step each sensor nodes have knowledge about the cluster number. According the Table 3.8 each sensor node have a table.

Node id	Position	Energy	Distance to sink/BS	Link	Hop size to sink/BS	Cluster ID
1	(75.85,45.48)	0.081	25.65	10	2	8
2	(28.24,63.59)	0.098	55.72	8	3	4
⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	(x_n, y_n)	E_n	$DtoBS_n$	$Link_n$	$Hop\ size_n$	$CluserID_n$

Table 3.8 Neighboring table for each sensor node

Each sensor node has knowledge about the own position, residual energy, distance to sink/BS, the number of neighbors (link), hop size to sink/BS and appropriate cluster-ID. Also, each node is aware of linked neighbors' position knowledge. According to Equation 3.11, the cost of each row based on node j calculated. For instance, cluster one that has twelve cluster members. For each of twelve members the population space declared and the cost computed. As the goal of these protocols to find the minimum cost of the cluster heads. The population agents try to find coefficients at a low cost.

$$f(i) = a_1E(i) + \ a_2L(i) + a_3Es(i) + a_4H(i) \tag{3.11}$$

According to the I-GWO and Ex-GWO algorithms structure, it will be different to find the optimal coefficients. In incremental grey wolf optimization (I-GWO) algorithm the first wolf or alpha wolf has the main responsibilities. If the alpha wolf finds a hunt in an optimal location, remaining wolves in the pack follow the alpha wolf's position. Otherwise, the alpha wolf predicts the hunt position in worst way, the reaming wolves in the pack move on the mistake position. In the expanded grey wolf optimization (Ex-GWO) the first three wolves' position experience has an effect on the remaining wolves in the pack. The fourth wolf, find the hunt position based on the first three wolves position also the fifth wolf finds the hunt position based on the first fourth wolves position. In this way, the hunt position information ahead of one wolf has a direct effect on the hunt finding position. By the way, the coefficients should be chosen from search space in the algorithm.

3.3.2.2. Cluster Head Selection in I-HEEL by GWO, I-GWO and Ex-GWO

I-GWO and Ex-GWO are explained in section 2.2.5. The cluster head selection mechanism is solved in the base station. Since the sink/BS energy is unlimited, the cluster head selection computations are done in the base station. After the confidence search space is determined in the previous section, here the optimal cluster in each round of the network will be determined. As the GWO, I-GWO and Ex-GWO structure is different, the following of this section is based on each of them.

3.3.2.2.1. Cluster Head Selection with GWO

In the GWO algorithm, there are four types of wolves' alpha (α), beta (β), delta (δ), and omega (ω) wolves. The alpha wolf is the leader of the pack and has responsibilities like, find the hunt and guide the pack wolves to hunt. The alpha wolves act as a leader in the pack, the alpha wolf's decision stronger than other wolves in the pack. The alpha wolf's decisions dictated to pack. Beta wolf act as co-leader in the pack, it is located in the second level of the hierarchy. Beta wolf is a suitable substitution for the alpha wolf. Delta wolf is in the third level of the hierarchy, this type of wolves follow the instruction that dictated by alpha and beta wolves. The delta wolf has an important role in the pack if in the pack there are not delta wolves. The pack encounters serious problems. In the last level of the hierarchy, the omega wolves located. The GWO algorithm based on these wolves and mimics the natural behaviors of the wolves in encircling, hunting and attacking.

In the I-HEEL, as mentioned previous section there is search space. The number of search agents is 15. The first wolf (alpha) consider the first row of the search space as the best solution. The beta and delta consider the second and third rows as the best second and third-best solutions. Then, according to Equation 2.26 (section 2.2) update the hunt position. Also, there are two random parameters to control the wolves encircle the hunt in the optimal position. After encircling, the hunting mechanism should be executed in the algorithm. The obtained position of each three wolf average is the next position of the fourth (omega) wolf's position. Then there is an attack from omega wolf to the hunt. There are $\vec{A}$ parameter that controlled by a value. If the $\vec{A}$ value is less than 1, the wolf should be attacking the prey, else try to search other prey (explore).

Then, according to the fitness function, the cost for each cluster member in a cluster calculated, the sensor node with minimum cost is chosen as the cluster head. Then, the base station broadcasts a message that contains the cluster head id to appropriate cluster members. The cluster setup phase in organized at the beginning of each round of the network. Then based on TDMA schedule the cluster head and cluster members transfer the data packets. The TDMA schedule avoids data collisions in the clusters.

3.3.2.2.2. Cluster Head Selection with I-GWO

I-GWO is an incremental version of the GWO algorithm. In I-GWO, the first hierarchy of the pack is the alpha wolf. The remaining wolves in the pack have the same responsibility. In other means, the alpha or leader wolf has the main responsibility of encircling the prey. The wolves in the pack must follow the alpha wolf's position to approach the hunt. In this method, each wolf in the pack except the alpha wolf, update own position based on the alpha wolf position. Authors in the I-GWO mentioned that I-GWO does not guarantee to find the best solution. Since the remaining wolves' behavior is based on the alpha wolf.

Also, in the I-GWO the number of search agents (wolves) is fifteen. The alpha wolf considers the first row in search space as the best solution. The second wolf, update the own position based on the alpha position to reach the hunt. The third wolf decides to move on to the next position based on the alpha and second wolves' position. The encircling, hunting and attacking in the I-GWO based on the equations on the section 2.2.5.1. After the one iteration is completed in the I-GWO the solution with minimum cost is chosen for the best solution. The best solution coefficients update dynamically in each iteration. The optimal sensor node with low cost is chosen for cluster head in the current round of the network. The selected cluster head has the maximum residual energy level and neighbor size and average energy of neighbors with minimum hop size to the sink/BS. Besides, the coefficients are not static, the algorithm described the value of each coefficient. In this way, the network will have a long network lifetime and consequently increase data packet delivery.

As mentioned before, the I-GWO algorithm calculation phases id done in the sink/BS. After the cluster head node is defined, the sink/BS broadcast a message to the sensor nodes. In other means, it is not broadcast, it is multicast because a group of the sensors (cluster) should be aware of appropriate cluster heads. In this way, the overhead of the sensor node is increased. Since the base station is calculated the cluster heads and there is only a message in each round. After this, the communication scheme between CH and cluster members and sink/BS is TDMA schedule to reduce collision in the WSN.

3.3.2.2.3. Cluster Head Selection with Ex-GWO

In the Ex-GWO, the expanded grey wolf optimization reaches the best solution. This algorithm working mechanism is different from GWO and I-GWO. In Ex-GWO the wolves hierarchy is defined as, alpha, beta, delta, and omegas wolves. It means the remaining wolves except the first three wolves are named as omegas. Here, the alpha, beta and delta wolves consider the first, second and third best solutions from search space. The fourth wolf, update the next position to hunt based on the first three wolves position. In other means, the wolf n update its own position based on n-1 of previous wolves. It means the n-1 wolves position experience is used for updating the position of wolf n. In our method, we consider 15 search agents (wolves). The 15th wolf updates its own position to reach hunt based on 14 of previous wolves. Ex-GWO execution time is slower than I-GWO. Furthermore, the computational complexity both of algorithms is $o(n^2)$.

Cluster head selection by Ex-GWO in I-HEEL is a novel method in cluster head selection. As mentioned before, the I-HHEL algorithm aims to increase the network lifetime. Since Ex-GWO tries to select the solution at a low cost. Each of the candidate sensor nodes examined in the Ex-GWO. In contrast the LEACH algorithm, there is not a set of sensor nodes that have been selected before as cluster head. The fitness function for every sensor node is computed. I-HEEL fitness function follows four parameters and coefficients. The coefficients are select dynamically. The encircling, attacking and hunting mechanism is to follow the equations on the section 2.2.5.2. In encircling the wolves pick the first, second and third solutions as the best solutions. In the attacking phase, the wolf n update the next position based on the first three wolves position. Then in the hunting, the |A|<1, the wolf attack the prey, otherwise search for other prey.

Node	**Residual Energy**	**Number of hops**	**Number of links to neighbors**	**Average Energy of node's neighbors**	
Node I	Ei	Hi	Li	Esi	Cost i
Node j	Ej	Hj	Lj	Esj	Cost j
⋮	⋮	⋮	⋮	⋮	⋮
Node n	En	Hn	Ln	Esn	Cost n

Minimum cost node selected as CH

Figure 3. 34 The cost table for cluster head selection

The selected cluster heads are the optimal and best sensor nodes from candidates. In I-HEEL, the cluster head numbers are static. As a result, the stability in the network increased. Connectivity in the I-HEEL is full connect in the network whole time. The selected cluster head is a node with a low cost to the sink/BS (Figure 3.34). In addition, according to the fitness function, the coefficients are dynamic. In each round, they updated according to the network situation. The cluster head operation is done in the base station. After this, the base station transfers the appropriate cluster head ID to the cluster members. The cluster heads transfer collected data packets from the cluster and transfer aggregated data to the sink/BS in single-hop communication.

Generally, the I-HEEL method finds the optimal cluster head for each round of the network. The metaheuristic algorithms applied to the I-HEEL algorithm to increase the network lifetime, number of alive sensor nodes, packet delivery. The proposed method reduces overhead in the network. As previously mentioned, the topology of the network is based on hierarchical topologies. The connectivity is this type of topology based on the cluster heads' availability. I-HEEL data transferring scheme follows the TDMA schedule. This avoids the data packet's collision. Our protocol does not consume more energy for cluster head selection since the cluster head selection is done in the base station. As each sensor nodes have a knowledge table of its own features and neighbors, the same as the base station has a data table of all sensors. It is clear that the HEEL and I-HEEL focus on four parameters in cluster head selection. We explain the I-HEEL performance by the simulation in the following section of this chapter.

3.3.3. Simulation and Algorithm Configuration

This section defines the network configuration parameters to simulation on the MATLAB software. I-HEEL algorithm is compared with the HEEL algorithm. I-HEEL algorithm runs with three different metaheuristic algorithms; GWO, I-GWO, and Ex-GWO. As to obtain the standard results, all of the algorithms are run in the same situations. The network configuration parameters outlined in Table 3.9. The metaheuristic algorithms parameters are the same for all algorithms. The maximum iteration for each algorithm is fixed on 100, the search agents are 30. The results of the simulation are discussed in the next section.

Table 3. 9 The configuration parameters for I-HEEL

Parameters	Values
Network Size	100*100 m
Number of nodes	100
Base station location	50*100 m
Data packet size	4000 bits
e_0	0.5 J
e_{fs}	10 pj/bit/m2
e_{mp}	0.0013 pj/bit/m4
e_{elec} (TX,RX)	50 n j/bit
Data Aggregation Energy cost	50 n j/bit

In addition, the network has some assumptions:

- The sensor nodes are immobile
- The sink/BS is settled in the center of the network
- The network has sixteen static clusters
- Each cluster has at least one sensor nodes
- The cluster head is responsible for data communication to the sink/BS
- The sensor nodes are deployed randomly
- All sensor nodes are homogeneous, fixed initial energy and transmission range

The network was performed in 100×100 meters. Also, all of the sensor nodes have leastwise one neighbor. The initialized network is stored to use in the other algorithms. Evaluation of the I-HEEL by simulation results on comparison metric parameters such, the number of alive nodes, the remaining residual energy, and packet delivery ratio obtained. According Equation 3.11 coefficients a_1, a_2, a_3, a_4 updates in each round of the network by metaheuristic algorithms. Contrary to the HEEL algorithm.

Also, in each round of network, the metaheuristic algorithms are determined the cluster heads. In the simulation software MATLAB, we consider node counts and sink/BS locations are static. As mentioned, the output metric parameters are the number of alive nodes, the average residual energy of the network, and packet delivery ratio. The following section describes widely, the obtained results.

3.3.4. Comparison and Results

The performance of the I-HEEL algorithm is evaluated in this section. I-HEEL is simulated in the MATLAB. The obtained performance of the I-HEEL is compared with the HEEL algorithm. The experimental results are obtained helps to analysis and compare the performance of the proposed method with other algorithms. As repeatedly mentioned both of the HEEL and of I-HEEL protocol is based on hierarchical protocols. First, Figure 3.35 describes as the number of alive nodes in the network of I-HEEL based on GWO, I-GWO, and Ex-GWO. As shown in the figure, the number of alive nodes in the I-HEEL based on the Ex-GWO is more than the other two algorithms. Respectively, I-HEEL based on GWO and I-GWO has the number of alive nodes. The lifetime on the I-HEEL based on the I-GWO is lower than two others, but in this method, the first dead nodes are on the 250th of the round, and the last dead is on the 383rd number of the round. As mentioned in the I-GWO algorithm description, the first wolf position has an effect on the remaining wolves in the pack. Obviously, this problem affected WSN performance in the number of alive nodes.

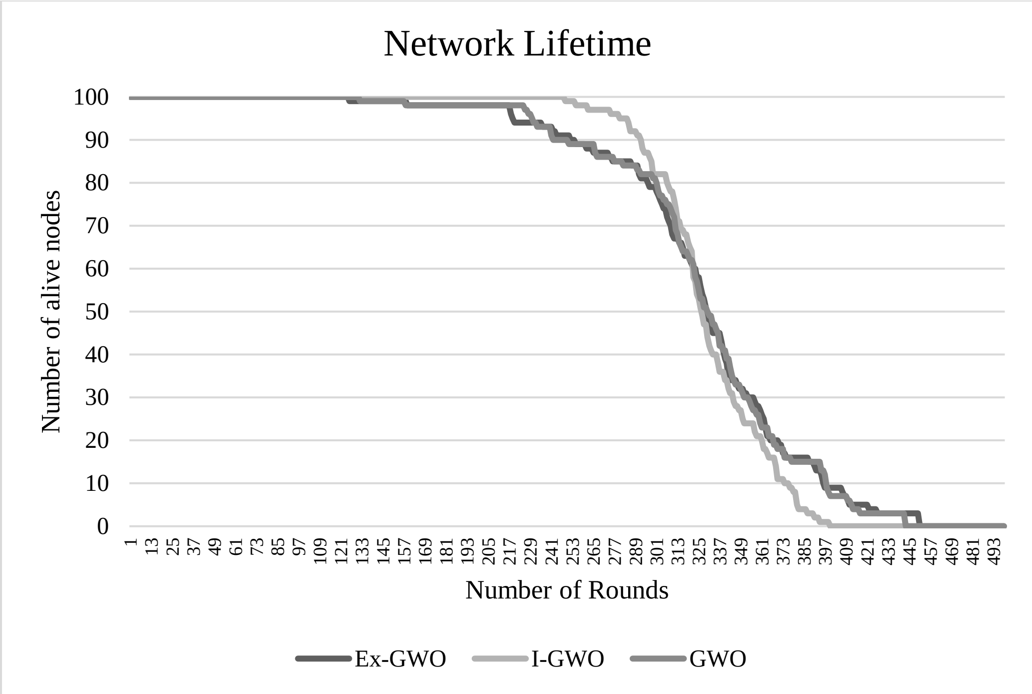


Figure 3. 35 The network lifetime of I-HEEL

In Figure 3.36, the HEEL algorithm is compared to the I-HEEL. Clearly, the metaheuristic algorithms improved the network's lifetime. As shown in Figure 3.36, the number of alive nodes in the HEEL algorithm is lower than the three other algorithms. In the presented figure, the results are very similar in the convergence curve. So, the obtained results compared with another way. In this comparison table, we define the parameters such as; the first node dies (FND), half node dies (HND), the last node dies (LND). According to Table 3.10, it is easy to evaluate the results.

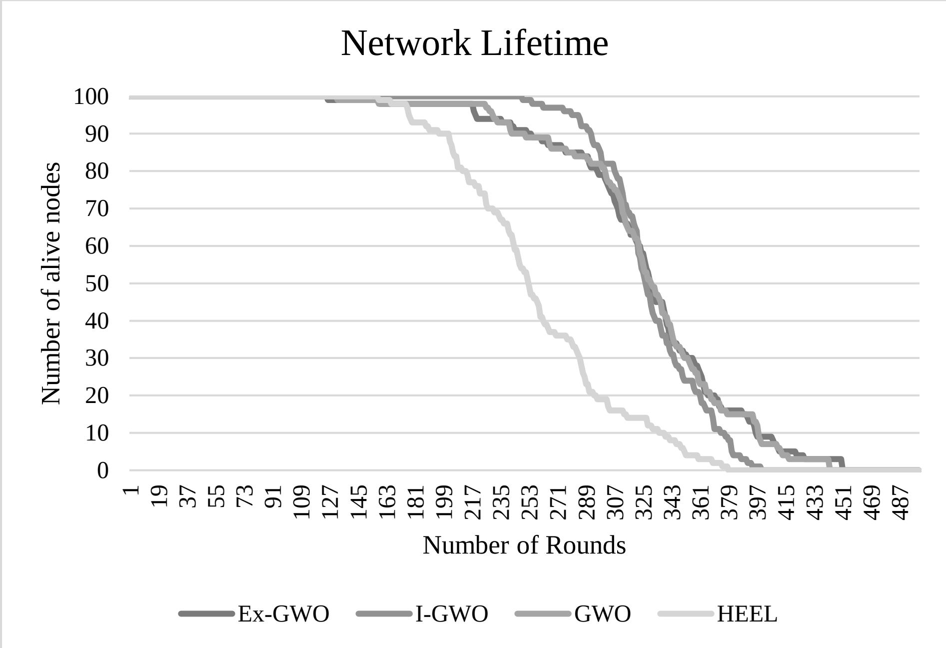


Figure 3. 36 The network lifetime of HEEL and I-HEEL

According to Table 3.10, the latest first node dies is on the I-HEEL method based on the I-GWO algorithm. The latest half node dies is on the I-HEEL based on the GWO and Ex-GWO algorithms. Finally, the important and the latest last node dies is in the I-HEEL based on the Ex-GWO algorithm in the round of 451. The whole network round is 500. I-HEEL method based on the Ex-GWO algorithm has the best performance in the number of alive nodes overall.

Table 3.10 The comparison of alive nodes

Method	First node die (FND)	Half node die (HND)	Last node die (LND)
HEEL	158	253	379
I-HEEL(GWO)	132	**330**	443
I-HEEL(I-GWO)	**249**	327	400
I-HEEL(Ex-GWO)	126	**330**	**451**

There is a significant improvement between HEEL and I-HEEL algorithms. The proposed algorithm has good performance in energy dissipation too. As in the presented figures, the convergence curves are in very close points, Table 3.11 outlined the obtained results. According to the table, the I-HEEL method based on the Ex-GWO has the longest residual energy network.

Table 3.11 The number of alive rounds on the HEEL and I-HEEL

Method name	Number of round	Ordering
HEEL	379	3
I-HEEL(GWO)	443	2
I-HEEL(I-GWO)	400	4
I-HEEL(Ex-GWO)	**451**	1

Figure 3.37 presents the percentage of successful data packets that the sink/BS is received. The packet delivery ratio is gain by Equation 3.12. I-HEEL method based on the Ex-GWO is obtained 88% of the data packets successfully, also the value of the others is 84%, 78%, and 76% respectively I-HEEL based GWO, I-HEEL based on the I-GWO and HEEL algorithms. This is just the number of successful data packets that sink/BS received. The number of lost packets calculated from the subtraction of all sensed data packets and the number of delivered packets.

$$packet\ delivery\ ratio\ = \frac{number\ of\ the\ packet\ that\ the\ sink\ received}{total\ of\ broadcasted\ packets} \quad (3.12)$$

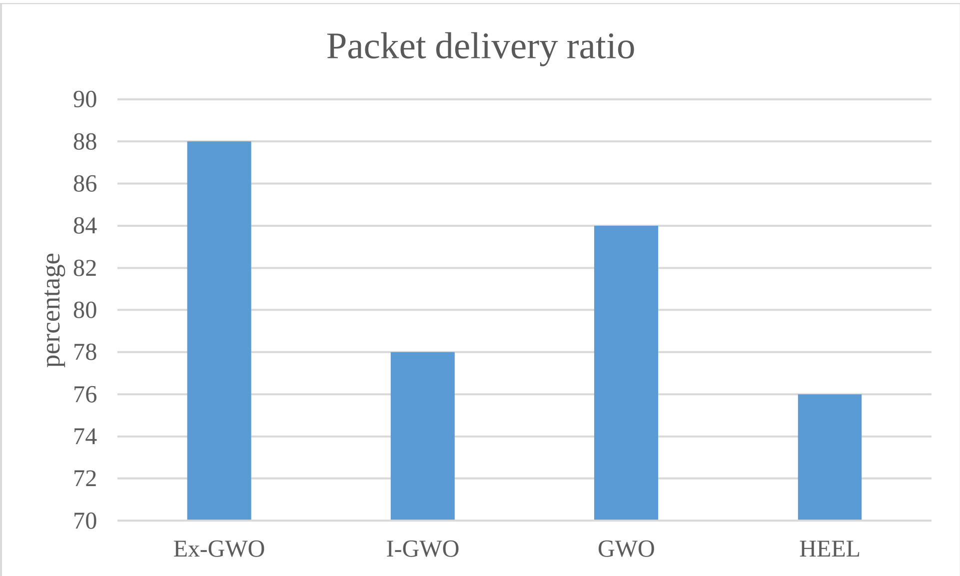


Figure 3. 37 Packet delivery ratio on the HEEL and I-HEEL

3.3.5. Conclusion and Outcome

I-HEEL is an improved version of the HEEL algorithm. HEEL algorithm is proposed in this thesis in section 3.1. I-HEEL is based on three metaheuristic algorithms to improve the performance of the HEEL algorithm. The metaheuristic algorithms that applied in the I-HEEL are GWO, I-GWO, and Ex-GWO. The last two algorithms also proposed by the author of the thesis. There is a search space as inputs in the metaheuristic algorithms that the search agents of the algorithm try to find the best solution. The studied metaheuristic algorithm in this chapter is on the swarm intelligence category. The grey wolf optimization (GWO) is imitating the social manners of the grey wolves in nature. Two other algorithms I-GWO and Ex-GWO are the modified version of the GWO algorithm in some habitats of the grey wolves. In GWO, there are three habitats of the grey wolves simulated such as encircling, hunting, and attacking. I-GWO and Ex-GWO have modified the encircling habitats of the grey wolves.

As in section 3.1 mentioned, the HEEL algorithm has fitness function to select the best and optimal cluster head in each round for sixteen clusters. The HEEL algorithm focused on the four main parameters of each WSN. The energy of each node, the average energy of the neighbors, hop size to the sink/BS, and the number of neighbors is the parameters in the fitness function. Besides, there is four coefficient for each of them. The coefficients are chosen static by simulation experiments. As a result, the HEEL method run with steady coefficients. After the simulation result and discuss on the HEEL algorithm, the author of the thesis improved the HEEL algorithm by three metaheuristic algorithms.

In I-HEEL the coefficients updated by three metaheuristic algorithms. The search space is initialized with the four random parameters for each search agent. Then, the GWO, I-GWO, and Ex-GWO find the optimal coefficients in each round of the network. In this way, the I-HEEL benefits the dynamic coefficients based on network situations.

I-HEEL algorithm is based on the hierarchical protocols. The sensor nodes are deployed randomly and the initial energy of all nodes is the same. The cluster head for each round is selected dynamic coefficients and metaheuristic algorithms. After the clustering scheme, the routing phase follows the TDMA schedule. The communication with cluster members and the cluster head also sink/BS are in TDMA schedule. In the next chapter of this chapter, the author proposed a new routing method based on the I-GWO and Ex-GWO algorithms.

The obtained results from the simulation in the MATLAB software presents valuable information about the proposed methods. I-HEEL improves the performance of the network in the number of alive nodes, residual energy and packet delivery ratio noticeable. In the number of alive sensor nodes, I-HEEL based on the Ex-GWO has the higher value between others. After I-HEEL based on the Ex-GWO, I-HEEL based on the GWO, I-HEEL based on the I-GWO and HEEL algorithms have the highest number of alive nodes respectively. It is necessary to mention that, the I-HEEL based on the I-GWO can continue the network until the 249^{th} of the network round. It means the proposed method is a good chose for sensor networks that the initial situation of the network is important. Since, until half of the network, the network connectivity is 100%. Although I-HEEL based on I-GWO has not a good performance in comparison of the Ex-GWO and GWO in the last node dies, the network connectivity until half of the network is important too. The WSN applications that need the longest network lifetime, the I-HEEL based on the Ex-GWO is a good choice.

Obviously, the residual energy of the network is also determined in the results. I-HEEL based on the Ex-GWO has the highest residual energy until the 451th round of the network. I-HEEL based on the GWO also has the second-highest residual energy. The proposed method, I-GWO, and Ex-GWO have a different encircling hunt; the hunt in the WSN is the optimal cluster head. The packet delivery ratio of the results determined that also I-HEEL based on the Ex-GWO has the best performance. Although the execution time of the I-GWO is faster than the Ex-GWO, the Ex-GWO has the best performance in the cluster head selection. It is mainly related to the algorithm's structure.

According to the simulation results, I-HEEL based on the Ex-GWO algorithm has the best performance. The performance is evaluated in the number of alive nodes, the residual energy of the network and packet delivery ratio. The algorithm reduces the lost packet numbers too. As the main goal of the thesis is to consume energy efficiently, the proposed method saves energy about the last iterations of the network. I-HEEL based on the Ex-GWO solved the energy problem of limited energy supply wireless sensor networks. The proposed clustering scheme also focuses on the main four parameters of each sensor network. In the following section, the author proposed a new routing algorithm based on the metaheuristic to evaluate the self-proposed metaheuristic in the routing. I-HEEL based on the Ex-GWO can be used in the network with high density.

3.4. Routing Algorithm based on tMetaheuristic algorithms

3.4.1. Background and Challenges

Wireless sensor networks (WSNs) are one of the subcategories of ad-hoc networks. WSN consist of many sensor nodes that deployed an area. WSNs have different applications of fields such as traffic monitoring (Seyyedabbasi et. al., 2016), agriculture (Kiani & Seyyedabbasi, 2018), automobiles (Tavares et. al., 2008), health monitoring (Kim et. al., 2006). Each sensor node transfer data packets to base station (BS) via single-hop or multi-hop. In WSN, sensor nodes gather data from the environment and send data to BS. BS collects all data packets and sends the collected data to a server for accessing the end-user. As sensor nodes work in collaboration, it is necessary to have a method in data transferring. Sensor nodes suffer a limited power battery. Also, sensor nodes have limitations in bandwidth, computational capacity, and memory space. It is a challenge to operate the complex computation in each sensor node. One of the advantages of the sensor node is easy to assemble in harsh environments. Consequently, it is difficult to recharge and change batteries because of the use of one-time batteries. Remarkably, the main issue in WSN is to increase the network lifetime.

Since network lifetime, directly in a deal with sensor nodes remaining battery level, energy consumption set to become a vital factor in wireless sensor networks (WSN). Most studies have investigated novel methods to balance energy resources. Network topology has an important role in network lifetime and availability. The clustering model is one of the schemes that improve the performance of the network (Elhoseny et. al., 2016). The main goal of the clustering strategy is to use resource energies efficiently. In clustering model, sensors are organized into clusters and logically it is a hierarchical strategy.

In this chapter, a new energy-efficient routing algorithm is introduced. The proposed method benefits metaheuristic algorithms I-GWO and Ex-GWO. Since the pathfinding in the WSN is the NP-hard problem, the metaheuristic algorithms try to solve it. In the proposed method, the network is flat and sensor nodes are deployed randomly. The parameters such as; sensor node residual energy, traffic, distance, buffer size and hop size are important in the next-hop selection. The metaheuristic algorithms determine the next-hop. The main purpose of the method is to minimize the traffic in the sensor nodes and increase network availability with improve fault tolerance. This method has improved by two metaheuristic algorithms incremental grey wolf optimization (I-GWO) (Seyyedabbasi & Kiani, 2019), and expanded grey wolf optimization (Ex-GWO) (Seyyedabbasi & Kiani, 2019). The fitness function for metaheuristic algorithms is based on five parameters.

3.4.2. Routing Algorithm Description

In this section, the new energy-efficient routing protocol based on the two metaheuristic algorithms explained in section 2.2.5 is presented. This method focused on the critic features of sensor nodes in the routing algorithms. Most of the routing algorithms based on residual energy of next hop. In this method, we focused on energy parameters in addition focused on traffic and buffer size. Also, the distance between the destination and sink/BS is important too. Besides, this method finds the best path from the possible paths. The best path has multiple hops. This feature is included in the network assumptions. The proposed method will be applied in the wireless sensor network application and Internet of things (IoT) projects that the network protocols are flat.

3.4.2.1. Definition and Designing Algorithm

In the network, sensor nodes are deployed randomly. The sink/BS is located inside of the network area. There are different paths between source and destination (base station). The proposed metaheuristic algorithms are based on the population. The search space of metaheuristic algorithms is a collection of randomly generated sensor nodes that correspond to a valid routing schedule. The search agents in the metaheuristic algorithm indicate the population size. In the I-GWO and Ex-GWO, the search agents' number is the number of grey wolves in the simulation. Therefore, the search space is a matrix that the rows represent the number of search agents and the column is the valid routing paths. The routing hops are different in the network. For example, in a network, the maximum number of hops between a sensor node and the base station is 5. Hence, there are different paths with 2, 3, 4 and 5 hops. For each number of hops, the method finds the best path. There is one best path. Then the optimal path is the minimum cost of the obtained best paths. I-GWO algorithm is based on the leader wolf behaviors. Other wolves in the pack update their own position based on all the wolves selected before it. In the Ex-GWO algorithm, the nth wolf updates its own position to the hunt according to the previous and the first three wolves. I-GWO algorithm is a fast algorithm in comparison with GWO and Ex-GWO algorithms.

In the proposed network, we consider each node information is saved in the sink/BS as outlined in Table 3.12. The nodes information obtained by a request data packet that sink/BS broadcasted in the initialization of the network. Residual energy, traffic status, buffer size, distance to the sink/BS, and neighbor list are stored in the base station. The optimal pathfinding is done in the sink/BS as it has unlimited energy supply. The search space is populated in the metaheuristic algorithms with the possible paths between the source node and the sink/BS. The assumption of the method is to find the optimal path. Consider the metaheuristic algorithm has 30 search agents; the search space is a matrix with 30 rows and hops number columns. The path with minimum cost is chosen as the best path in each round of the network. The fitness function is determined the cost of each path from search space. Equation 3.13 gives the fitness function.

$$F_{i,j} = \left(c_1 d_{i,j}\right) + \left(c_2 H_j\right) + \left(c_3 \frac{\text{Valid}-\text{Traffic}}{\text{T}_{i,j}}\right) + \left(c_4 \frac{E_{initial}}{E_j} . \frac{\text{Buffer Capisity}}{\text{B}_\text{j}}\right) \tag{3.13}$$

Equation 3.13 calculates the cost between two nodes *i* and j. where E_j indicates the residual energy of the node j, H_j is the hop size between nodes j and the sink/BS. $D_{i,j}$ is the distance between nodes i and j. $T_{i,j}$ is the traffic status between nodes i and j. B_j Indicates the buffer rate of the nodes j. $c1, c2, c3, c4$ are four control parameters between 0 and 1 where $c1 + c2 + c3 + c4 = 1$ where $\text{c1} < \text{c2} < \text{c3} < \text{c4}$. Metaheuristic algorithms (I-GWO and Ex-GWO) also calculate these control parameters. The coefficients updates are also done in each round of the network. This mechanism is used in the I-HEEL algorithm. First, for each possible path with n hops the costs calculated, this process is performed for all paths with different paths. Then, the minimum cost is chosen as an optimal path. In this method, there is a dynamic hop number path. For instance, in the first round of the network may be the hop size of the path is 5, in other iteration the hop size number is 3.

Table 3.12 The information table of each node (store in the sink)

Node ID	Energy	Traffic	Buffer	Hop size to BS	Neighbors list
i	0.5	1	1	4	J, k
j	0.5	1	1	3	I, k

The cost between two nodes calculated until reach the sink/BS. Here, in the path, the packet is passed just one time from each sensor node. These costs are collected then the sink/BS is choosing the path with minimum cost (Equation 3.14). After all, the sink/BS broadcast routing path to source nodes. For instance, in the search space, the candidate path is (N4, N61, N98, N43, and BS). First, the cost of tuples (N4, N61), (N61, N98), (N98, N43), and (N43, BS) are calculated. The collected value is the cost of the path. After, all candidate paths are calculated the sink/BS decides to choose the minimum cost value (Table 3.13).

$$Cost_{S,D} = \sum_{i=1}^{j=n} cost_{i,j} \tag{3.14}$$

Table 3.13 At the end of each iteration candidate paths cost

Candidate paths	Value
Path 1	0.78
Path 2	1.36
⋮	⋮
Path n	n

The cost of the two nodes is related to the global knowledge of two sensor nodes. As mentioned, the optimal pathfinding is done in the base station. Also, there is a table as nodes information in the sink/BS. In this table, some basic information stored such as; each node residual energy, traffic status, buffer rate, hop size to sink/BS and neighbor list. Each sensor node has a routing table stored in its memory (Table 3.14). In this table, neighbors list, distance to neighbors, distance to sink/BS, and hop count to sink/BS are stored.

As seen in the Figure3.38, in the first step, there are search spaces of the path with the various numbers of hops. Here, the metaheuristic algorithms find one best path for each search space. The obtained path has minimum cost on that search space. In the second step, the algorithm chooses these paths. In the third step, the algorithm selects the minimum cost path from obtained paths, as an optimal path between source and destination.

Table 3. 14 Routing table for node i (store in the node)

Neighbor lists	Distance	Distance to $_{BS}$	Hop count to BS
i	--	18	4
j	15	12	3

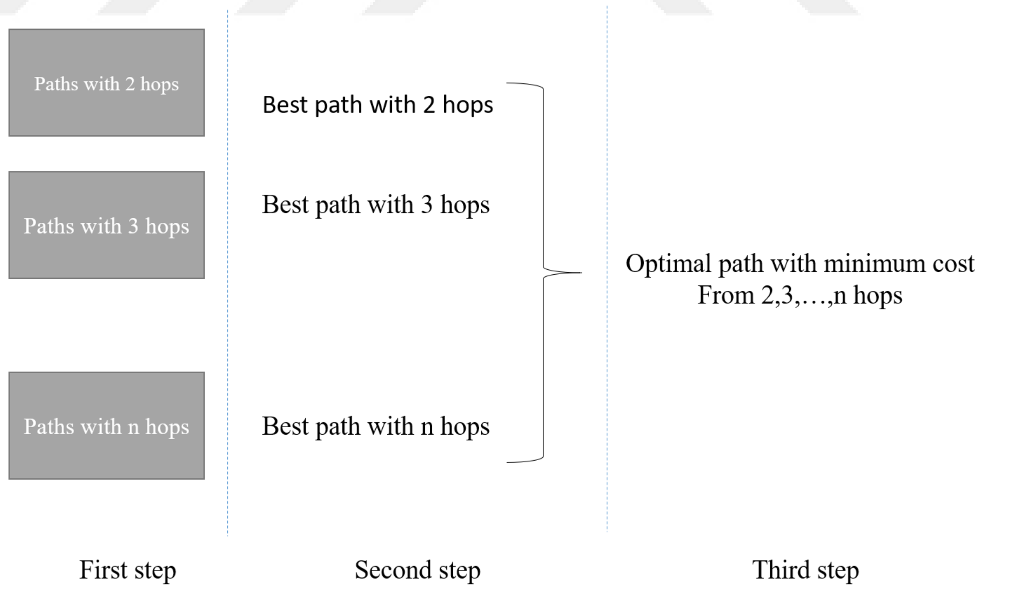


Figure 3. 38 The routing method architecture for finding optimal path

3.4.2.2. Definition Packet Frame

The data packets that broadcasted from sink/BS and sensor nodes have specific frames to transfer data between each other to choose the best path. In the initialization of the network, the sink/BS broadcast a message to get the global information about the sensor, also this message broadcasts at the beginning of each round (Figure 3.39). The sensor nodes answer the sink/BS data packet and transmit information consists of residual energy, traffic status, buffer rate, hop size to sink/BS, and neighbor list (Figure 3.40).

Source ID	Request data	CRC

Figure 3. 39 The data packet that BS asked

Source address	Destination address	Residual energy	Traffic status	Buffer ratio	Hop size to BS	Neighbors list	TTL

Figure 3. 40 The data packet that sensor answered

After, this global information helps the sink/BS to find the optimal path from a source to the destination (sink/BS). This operation is done in there since the sink/BS has unlimited energy supply and can compute the complex operations too. Then, the base station finds the minimum cost of the path, so the sensor node is aware of the routing path. In this way, the base station broadcasts a packet contains the source and destination ID, path sensor nodes (Figure 3.41).

Source ID	Destination address	Path	Data	Options	Seq

(Source ID, $N_{54}, N_{17}, N_{87}, BS$)

Figure 3. 41 The data packet that BS broadcast after find the optimal path

3.4.2.3. Computes parameters

As mentioned, the data packet that each node answer to the sink/BS request contains some parameters. The residual energy of each node is the remaining battery level of the sensor node. This parameter included in the data packet as a Joule metric. Traffic status is obtained using the valid traffic divided by traffic between nodes i and j. The valid traffic is defined in network assumptions. Buffer ratio is also calculated by the buffer capacity of each node divided by the node j current buffer size. Each node when getting a broadcasting message based on the distance between the sensor node and sink/BS calculates hop size to sink/BS. Besides, each node should find the neighbors, by handshaking mechanism (Guha et. al., 2008).

3.4.2.4. Fault tolerance

The routing step to the path is defined by the base station. In the routing algorithms, one of the challenges is to control the sensor node's failure. For example, the routing path is defined, but in the routing operation one of the sensor nodes along the path is unavailable (battery drains) the routing should change to the path. The robustness and fault tolerance is important. Also, the seq frame checks the packet duplication in the sensor nodes. Each sensor nodes, first check the seq frame if the receive the packet previously, immediately, discard the packet. In the proposed method, as all of the sensor nodes, neighbors list and distance is stored in the memory. The algorithm tries to find an alternative sensor node. In this way, some factors like energy, traffic, and buffer have been ignored, just the distance and hop size is important. Since the main goal is to transfer the data packet to the sink/BS. In this way, the sensor node i that should transfer data packet to node j, and node j is unavailable. Node i looking for its routing table and chose a neighbor (node k) that have minimum hop size to the sink/BS (Figure 3.42). So, the node i calculate the cost between node i and k and add the cost in the options frame of the data packet (Equation 3.15).

$$Cost_{S,D} = \sum_{i=1}^{j=n} cost_{i,j} + \sum_{i=1}^{k=n-j} cost_{i,k} \qquad (3.15)$$

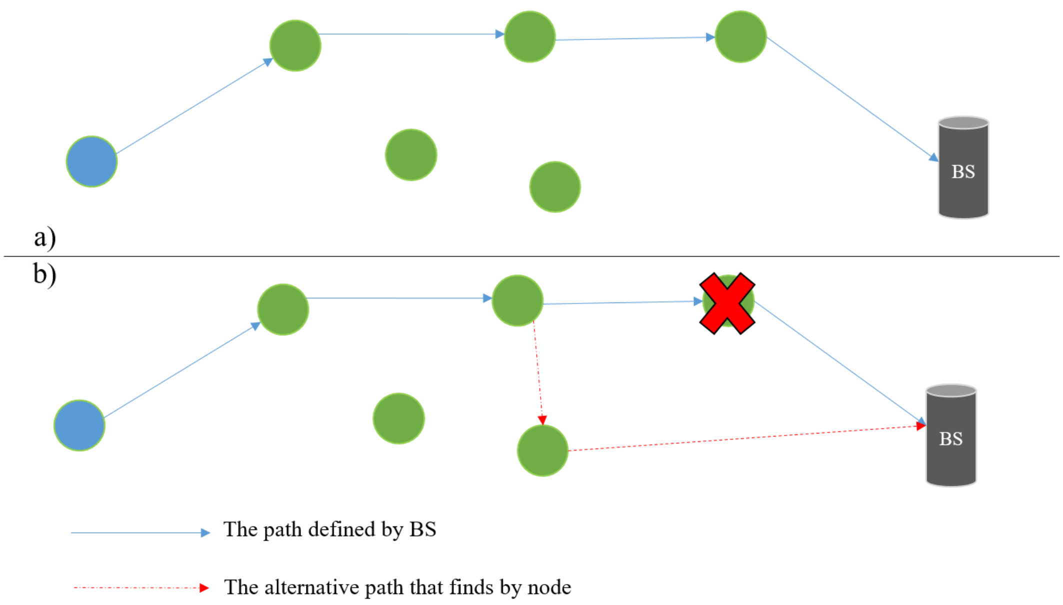


Figure 3. 42 Fault tolerance in routing algorithm

3.4.2.5. The network round and metaheuristic iteration

In the actual application of sensor networks, the network round and metaheuristic algorithms iteration that worked separately. In the network considerations, the rounds and iterations should be considered separately. Each round of the network occurred in certain time intervals. In the proposed method, we consider a time interval between each round of the network. If the round and iteration are worked together it reaches to overload on the network. Metaheuristic algorithms find the best solution. On the other hand, there is not a specific time to reach the solution, in this way; we do the pathfinding operation in the base station. After this, in a period, the network should be finished the data packet data transferring from destination to the source. As two metaheuristics used in this method (I-GWO and Ex-GWO), they have different execution times. According to the simulation results on the benchmark (Seyyedabbasi & Kiani, 2019), the I-GWO execution time is faster than the Ex-GWO algorithm. Both algorithms based on the grey wolves' habitats of hunting and foraging and survival behaviors. Although the main algorithm of GWO is proposed as the same habitats, the I-GWO and Ex-GWO modified the encircling hunt mechanism to find the best solutions. The following section is discussed the network model and simulation parameters.

3.4.3. Simulation Parameters

This section discusses the simulation configuration of the proposed method that simulated in the MATLAB software. The proposed method is an energy-efficient routing algorithm that applied in the two metaheuristic algorithms I-GWO and Ex-GWO. The network model is used in this method is flat. All the sensor nodes are deployed randomly in an area. The proposed method compared with GAR (Gupta et. al., 2013), BEE (Ari et. al., 2016), ACOHCM (Jiang & Zheng, 2018). The network input configuration parameters are the same for all algorithms (Table 3.15).

Table 3.15 Input parameters for routing algorithm based I-GWO and Ex-GWO

Parameters	Values
Network Size	100*100 m
Number of nodes	100
Base station location	50*100 m
Data packet size	4000 bits
e_0	0.5 J
e_{fs}	10 pj/bit/m2
e_{mp}	0.0013 pj/bit/m4
e_{elec} (TX,RX)	50 n j/bit
Data Aggregation Energy cost	50 n j/bit

Also, the sensor nodes in the network are static and deployed randomly in a 100×100 meters area. The base station is in the center of the network. All sensor nodes have at least one neighbor. Each sensor can transfer the data packet to the sink/BS. The maximum hop size to transfer a data packet from the destination to the base station is four. All sensor nodes are homogeneous; they have the same initial energy level (0.5 Joule) and communication range. Besides, the network has 500 rounds, that each round duration time is 2 seconds. The metaheuristic algorithm parameters are also important. I-GWO and Ex-GWO have 30 search agents (population size). The algorithms maximum iteration is 100. In the GAR algorithm, the initial population considered 300 chromosomes. In the crossover operation, 5% of the best chromosomes are selected by the tournament selection procedure. Also, the maximum iteration for the GA algorithm is 100. In the routing algorithm with the BEE algorithm, the population size is considered 30. The limit for neighborhood search is 20. Also, the iteration maximum size in 100. In the ACOHCM method, the number of ants (population size) is considered 30. The ACO algorithm iteration number is 100.

The metrics for performance evaluation are a network lifetime since network life is an important challenge in wireless sensor networks. In addition, the packet delivery ratio, as the number of the data packets that the base station received, and lost data packets can clarify the performance of the routing algorithm. Furthermore, the first node dead in the network is simulated to evaluate the network with this metric. The following section is discussed in the obtained results of the simulation.

3.4.4. Comparison and Results of Routing Algorithm

In this section globally described the evaluation of metrics on the proposed method and other algorithms. All algorithms compared in the same network input parameters. The routing algorithm based on the I-GWO and Ex-GWO algorithms is compared with GAR (Gupta et. al., 2013), BEE (Ari et. al., 2016), ACOHCM (Jiang & Zheng, 2018). The proposed method is applied with two metaheuristic algorithms, that the performance evaluation of each one is compared. Although the obtained results are nearly the same, the structure of metaheuristics is different from each other. Both the structure and hunting mechanism in the pack is separate. The grey wolves in the I-GWO update their own position to move on the hunt based on the alpha and previous wolves' position. In the Ex-GWO, the remaining wolves to update own position based on the first three wolves in the pack and other wolves before. In this way, the hunting (encircling) mechanism is different.

Figures 3.43 and 3.44 present the residual energy of the network during simulation rounds. This metric analyzes the remaining energy. The obtained results show that both algorithms have near the same energy consumption. However, the energy consumption the routing algorithm based on the Ex-GWO is less than the other method. Both of the routing and transferring methods are the same but the metaheuristic structure is different.

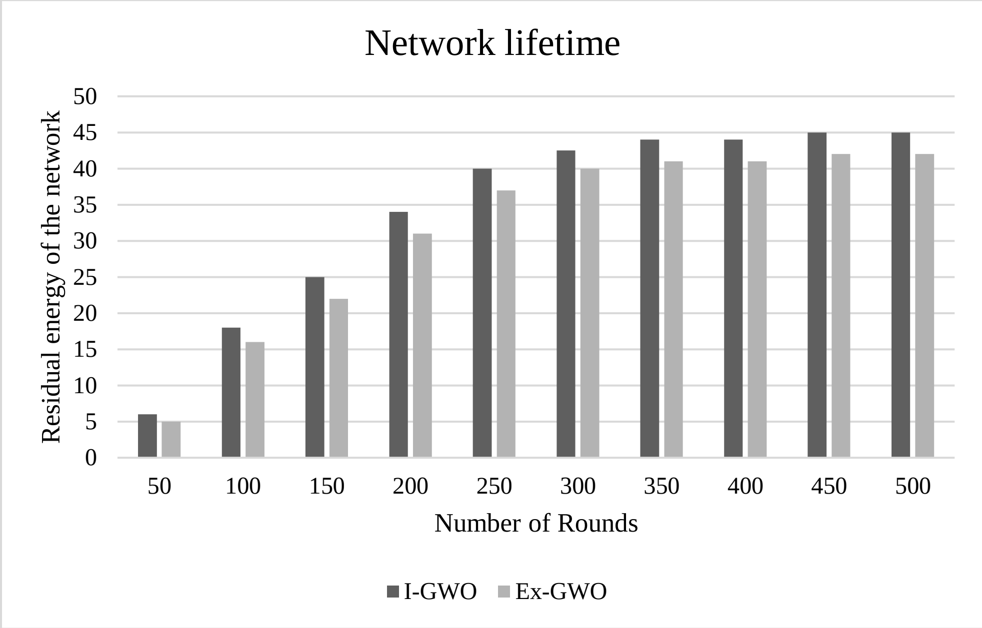


Figure 3. 43 Residual energy of the network based on the I-GWO and Ex-GWO

In Figure 3.44 the routing algorithms based on the I-GWO and Ex-GWO with other algorithms residual energy are compared. As seen, the proposed method can save energy even in the last rounds of the network. If this network continues until 700 rounds, the proposed method maybe continue too. The consumed energy in the network is a challenge. Network connectivity and data are strongly based on the residual energy of the sensor nodes. Also in the flat-based networks, each of sensors energy level have an effect on the others. In this type of network, there are not cluster head nodes to transfer collected data from clusters to the sink/BS. So, in the flat-based network, any sensor nodes' residual energy is important to network performance.

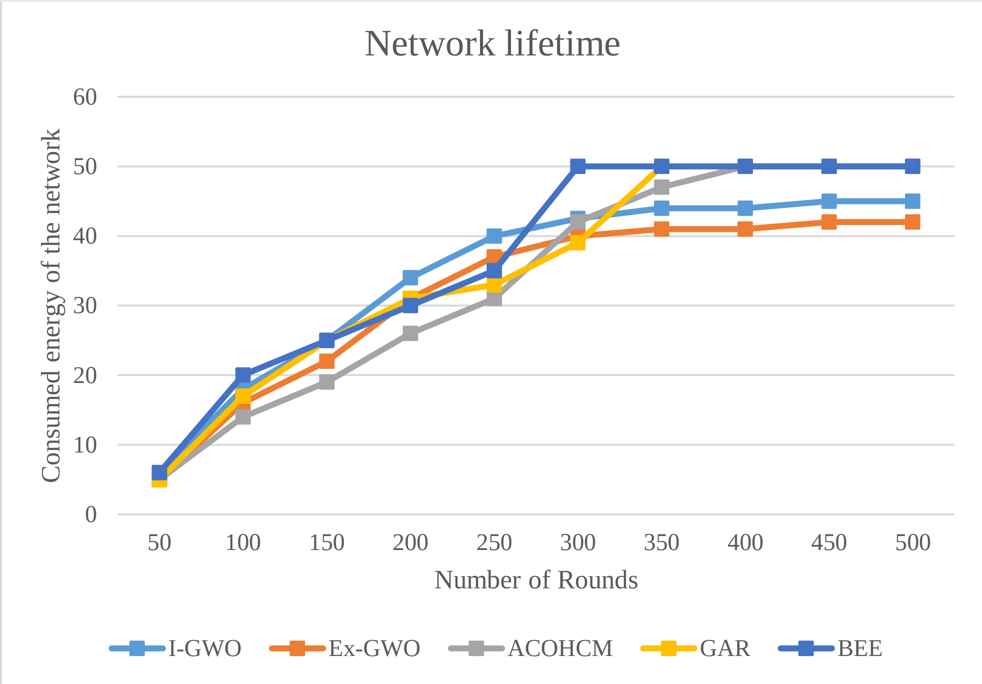


Figure 3. 44 The consumed energy in routing algorithms based on the I-GWO and Ex-GWO

Figure 3.45 shows the number of alive nodes in each round of the network. The number of alive nodes is illustrated, the number of nodes that have energy in rounds. The nodes with drained energy named as dead nodes. The proposed method to improve energy consumption, in the alive number of nodes, has good performance too. The other algorithm that applied with the genetic algorithm, ant colony optimization, and bee colony optimization have different alive sensor nodes number. As in the ant colony optimization routing algorithm, as the ACO continues, the residual energy of the network moves in a steady way. Since the pheromone, value is affected by the pathfinding.

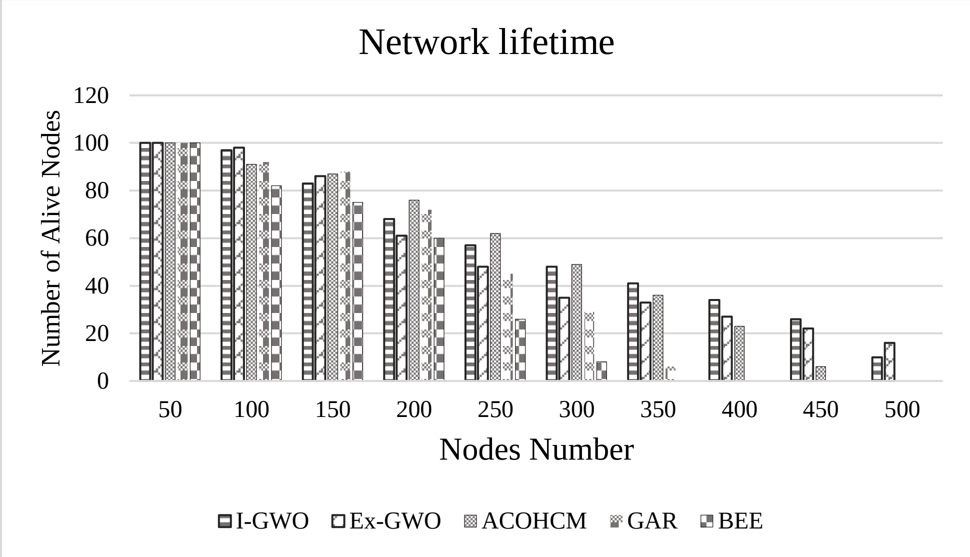


Figure 3. 45 The number of alive nodes in routing algorithms

The throughput of the network analyzed by the number of data packets to the base station. The amount of the delivered packets and packet loss rate gives the throughput evaluation of the routing algorithm. The sum of the received data packets by the BS divided by the sum of the send data packets to the BS gets the successful delivery ratio. The difference between the successful delivery ratio and 1 is the lost packet rate. The following Table 3.16 presents the amount of successful data packets sent to the sink/BS. As shown in Figure 3.47 the routing algorithm based on the I-GWO and Ex-GWO the successful packet ratio is obtained.

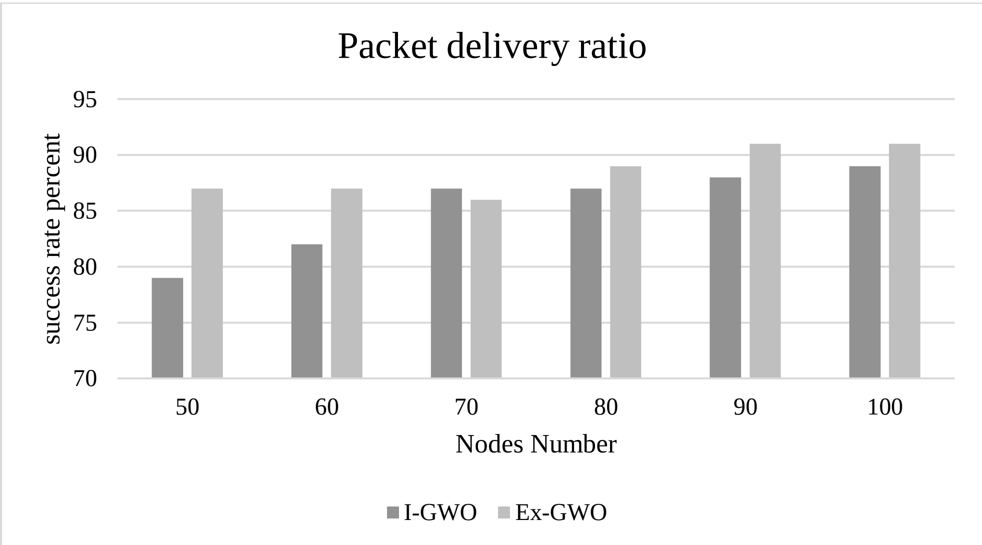


Figure 3. 46 The successful packet that base station is received

Table 3.16 Obtained packet delivery ratio in different number of sensor nodes

Sensor node number	I-GWO	Ex-GWO
50	79	87
60	82	87
70	87	86
80	87	89
90	88	91
100	89	91

Figure 3.47 presents the successful delivery packet to evaluate the throughput between all algorithms. The proposed method has a good delivery packet ratio in comparison to others. As the structure of each of metaheuristic, also, the routing method is different; the obtained value is lower than our method. If the same type of metaheuristic algorithm with different routing methods is compared, may be the comparison results are near to each other. Unfortunately, GWO algorithms do not apply in the routing algorithm until now. Since this thesis used other metaheuristics to comparison and evaluation.

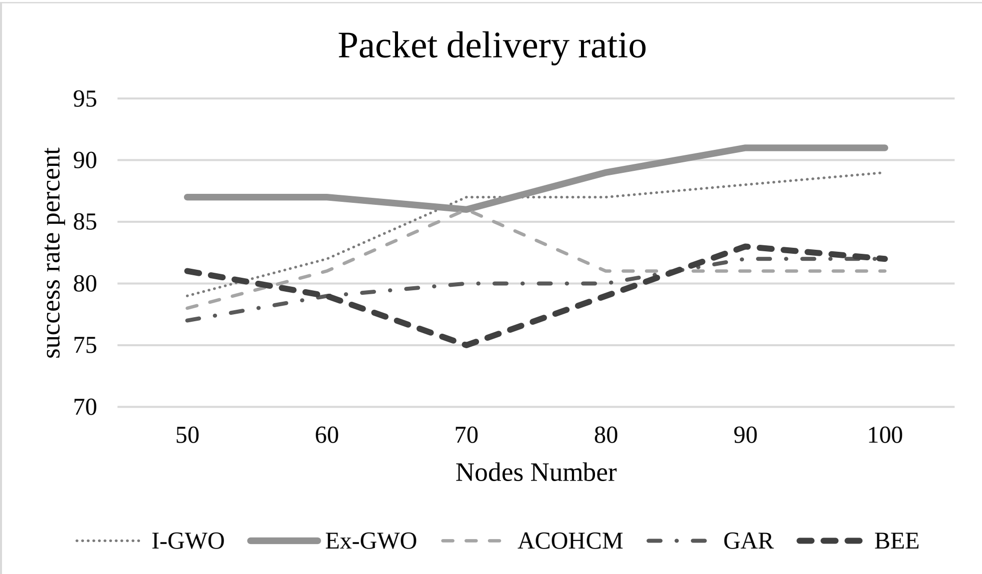


Figure 3. 47 Packet delivery ratio in routing algorithm

3.4.5. Conclusion and Outlook

The improvement of the network in the energy consumption is one of the active challenges in the wireless sensor networks. Since the sensor nodes suffer from limited energy supply. There are different schemes to improve network consuming energy. Routing algorithms, cluster method, data aggregation are the schemes that try to enhance the performance of the network. The proposed method in this section is an energy-efficient routing algorithm to find the optimal path. The proposed method is flat-based network. In this type of network, all sensors have the same and similar roles. The routing performs in multi-hop to transfer data packets to sink/BS. The routing start from the sink/BS queries to collect data. The metaheuristic algorithms applied in the routing method to find a path. The metaheuristic algorithms find the best solution from a search space. Since the sensor nodes' energy supply is limited, the finding optimal path is performed in the sink/BS. The sink/BS benefits unlimited energy.

The proposed method is a metaheuristic algorithm-based energy-efficient routing algorithm for wireless sensor networks. The algorithm has been provided with the grey wolf optimizer mathematical behaviors. The habitats of grey wolves are simulated such as encircling, hunting, and attacking the hunt. Two metaheuristic algorithms I-GWO and EX-GWO have been modified in the encircling mechanism. The population of the metaheuristics is considered the possible paths between nodes i and j. The maximum hop size for each path is x. It is caused all of the possible paths are be the same hop. The goal of metaheuristic algorithms (I-GWO and EX-GWO) to find a path with a minimum cost. The fitness function computes the cost of each path. Each metaheuristic has a maximum iteration number. After iterations, the best solution is the best optimal path for the network in the current round.

The routing operations performed in the base station. The fitness function that determines the cost of each path consists of some parameters. Consider node i transfer a data packet to node j. in this way, node i first check the features of node j. the residual energy of node j is important. Hop size of node j to the sink/BS is supposed. The distance between nodes i and j. the traffic status between nodes i and j. Furthermore, the buffer size of the node j has an impact on the fitness function value. Also, there are two random control parameters. The base station, for each candidate path, cost the fitness function between two nodes. As mentioned, the maximum hop size should be x. consider x is four; the fitness function performs this operation in x+1 time. After finding the cost value for each tuple, all the values a collected and the cost of that candidate path is obtained. Each of the nodes has a routing table that keeps the neighbors' lists, the distance between two nodes, distance to the base station and hops size to the base station. Besides the global information of each node has been in the base station by inquiry message that broadcasted from BS in the initialization of the network. The information of each node such as; residual energy, traffic status, buffer size, hop size to sink/BS, and neighbors lists. These tables help the sink/BS to find the optimal path in the network.

There are different data packets in the network to collect information and transfer information. After the optimal path is chosen by the sink/BS. A data packet that contains the source and destination address, path, data, seq, and options frame. The path frame contains the sensor nodes to identify numbers along with the routing. Data is collected information from sensor nodes. Seq is the sequence number of the pack to avoid the packet duplication. The options frame controls fault tolerance. When a sensor node in the path is unavailable, the method changes to the path to other sensor nodes by calculating the cost. The obtained cost value is added to the options frame. In addition, the options frame if it has a value, the base station is aware of the sensor node's failure in the path. This helps the base station to not consider that node in the next pathfinding. Each round of network is working separately from the iterations of metaheuristic algorithms. It is not the same. Each round has an interval time intervals. We considered two seconds for each time interval. The iterations of the metaheuristics are separate and in this method are 100 for each algorithm. There is not a time period in these algorithms since they are not time-dependent. Find the best solution (path) is important for algorithms. I-GWO algorithm has a faster time execution time than the Ex-GWO algorithm.

According to the simulation results obtained from the MATLAB software, the proposed methods based on the I-GWO and Ex-GWO has better performance. The network lifetime metric is evaluated in two ways. The number of alive sensor nodes and the total residual energy in the network. Both metric results prove that the proposed method enhances network performance. Besides, the routing method based on the Ex-GWO algorithm has better performance than the routing method based on the I-GWO. Although the execution time of Ex-GWO is long, they found the best paths. The better efficiency of the Ex-GWO has related to the structure of this algorithm since they used all the search agents' knowledge in the pack to find the best path. It is proved that swarm intelligence is stronger than particle intelligence. While, the I-GWO is not particle intelligence, but they benefit the first (leader) search agent (wolf) knowledge to move on the solution (hunt). In the data packet delivery metric, the proposed methods are also evaluated. The routing algorithm based on the Ex-GWO is better than others are. It is related to the network lifetime too. Generally, the routing based on the Ex-GWO is good choice for wireless sensor networks. Furthermore, the proposed method is suitable for large-scale networks.

3.5. Concurrent Path Finding in Real-Time Wireless Sensor Networks by a New Routing Protocol Inspired by Ant Colony Optimization

3.5.1. Background and Challenges

Wireless Sensor Networks are a family of wireless networks that do not need access points. They are one of the Ad-hoc network categories. In these networks, low resource devices are used; therefore, efficient resource consumption is always a priority in the design of the networks. Due to the sensor nodes' unique characteristics and flexibility with different technology substructures, these networks can be applied in various fields (such as military, agriculture, health, animal monitoring, etc.). In addition to IoT-based technologies, which are very popular in recent years, they have been used in other branches of Ad-hoc networks such as VANET and FANET. In the literature, many studies have been carried out in recent years in order to manage these resources (with priority energy and memory). All the studies have worked to manage resources in a good way based on networks for different purposes. Here, the metaheuristic algorithms have good role in designing routing algorithms. This method aimed to make efficient solutions inspired by these algorithms.

Optimization methods and algorithms are categorized into two groups of exact and approximate algorithms. Exact algorithms are capable of finding the optimal solution in a precise manner, but they are not efficient enough for difficult or hard optimization problems and their execution time expands exponentially with the dimensions of the problems. Approximate algorithms are capable of finding good (near-optimal) solutions in a short time for difficult/hard optimization problems. Approximate algorithms are also classified into three categories: heuristic, metaheuristic, and hyper-heuristic algorithms. Therefore, fractal or metaheuristic algorithms are a kind of random algorithm used to find optimal solutions. In metaheuristic algorithms, the path to reaching a solution is not important to us, the only point to consider is to find a solution to the problem.

The main goals of the metaheuristic algorithms are to prevent possible solutions from being captured at local optimum points as well as early convergence. The classification of metaheuristic algorithms can be applied based on various criteria such as single or multi (population) based solutions. Solution improvement in single-based methods are done during iterations, but the population-based methods perform optimization using a set of solutions. One of the branches of this category is Swarm Intelligence (SI) methods. Metaheuristics may be classified into three main classes as evolutionary, physics-based, and SI algorithms. As the ACO algorithm is in the swarm intelligence algorithm, we described this category.

The working mechanism in swarm intelligence (SI) methods generally based on nature-inspired and is based on a herd or collective social behaviors and community mind. Partial Swarm Optimization (PSO) (Eberhart & Kennedy, 1995), Ant Colony Optimization (ACO) (Dorigo & Di Caro, 1999), Artificial Bee Colony (ABC) (Karaboga & Basturk, 2008), Grey Wolf Optimization (GWO) (Mirjalili et. al., 2014) algorithms can be mentioned in the studies in this field. SI-based methods can solve complex problems more efficiently. However, it was mentioned before, it may not always guarantee optimized solutions. Although, the possible differences in the metaheuristic methods, all of them have a common concept. In all of them, the search area is divided into exploration and exploitation phases. In the first phase, the environment is investigated and the other phase uses the result of the first phase. It is necessary to mention that metaheuristic methods may not always be good for solving all problems.

SI-based methods can also be used in the decentralized and self-organization distributed systems so it is making them a good choice for WSNs. In particular, it may be a useful solution to the routing mechanism of the networks. Because finding the number of possible routes (solutions) is from the complex and non-deterministic polynomial problem family. In the proposed method, we applied a new ACO algorithm in the WSN. Also, the ACO algorithm is explained in section 2.2.1.

In the literature, many studies have been done for different purposes of WSNs by using the ACO method. However, many of them do not focus on full-size and efficient resource consumption. In addition, many of the proposed algorithms are not more suitable for the actual applications of WSNs and have been developed with too many assumptions. On the other hand, in most studies, no clear information is given about the working mechanism of the proposed algorithms (such as sequential, parallel and concurrently). Since there are more than one starting and ending points in the real field, it is correct to look at the concurrent and even parallel operating mechanisms. While there is no clear information about whether the defined ants are all started from the same point or they start from different nodes and there also are not scenarios with multiple sources and destination nodes.

In this section, a new ACO algorithm, which is useful for multipurpose and efficient resource consuming in WSNs, has been proposed. In this section, a protocol is proposed to real-time data transferring and pathfinding for series, concurrent and parallel possible structures on multiple sources, destination points (nodes) based on design factors in the actual application areas of WSNs. A new algorithm has been proposed for the routing mechanism of this protocol that is inspired by the ACO algorithm. In this protocol, the network is flat, path establishment method is a hybrid and next-hop selection based on the probabilistic method. There are some protocols in WSN that inspired by the ACO to reduce energy consumption. The well-known of these algorithms such as; ACO router chip (Okdem et. al., 2009), ACOHCM (Jiang & Zheng, 2018), LTAWSN (Mohajerani & Gharavian, 2016), EEABR (Camilo et. al., 2006) and other well-known algorithms explained in the related works of the thesis (Section2.3). The following section is described a new metaheuristic method inspired on ACO to efficient routing and resource consumption in multiple sources and destination models.

3.5.2. Concurrent Path Finding Method

This section is described as an efficient routing model with balanced consumption resources in the whole of the network. Therefore, this method not only chooses the path according to some parameters such as energy or short paths but also pays attention to traffic density on the routes, queue availability of the nodes and data reliability. In order to achieve these goals, it is careful to use balanced resources. In this method, because it finds routes between multiple sources and target nodes, it is more complex and also complete than many studies in the literature. The proposed method is designed for the WSN structure and related targets, inspired by the ACO. Therefore, the new version of the metaheuristic method based on ACO has been developed.

3.5.2.1. Definition and Designing Algorithm

In this method, packets sent in the network are called artificial ants as they go and come back from each source node to destination node on paths (edges). Each packet is an ant. In other words, each ant carries one packet. Each ant leaves the pheromone while crossing the path it chooses. These pheromones have an effective role in the choice of paths of other ants. The ants' path is chosen not just pheromone ratios, but the energy of the nodes, road traffic, and so on parameters. Two functions are defined for it that one of them is the heuristic function and the other is the pheromone function. In the ACO algorithm, the values of these two functions are kept in two separate matrices (tau and heuristic matrixes). This mechanism is not strictly logical to use in some structures such as WSNs. Because these matrices should be kept and updated separately by all sensor nodes.

This is not possible with respect to memory and processor power of them. In addition, significant delays and data loss may occur within the network. Furthermore, it is not possible to implement the classical ACO algorithm due to the dynamic nature of these networks (the change of the energies of the nodes, the state of the traffics on the paths, the change of the density of the package at the time dimension, congestion state, buffer size and availability rate of it and so on). Indeed, the value of both matrices is constantly changing. Therefore, a more accurate solution may occur if each node holds only the information of its neighbors in its routing table.

In this protocol, many issues such as how the routing table of each node should be, the definition of the contents of the ant packets, the mechanism of discarding of the repeated packets as so on are designed. The main goal of the proposed algorithm is to find the most efficient path between S-D nodes. However, the founded path(s) in the network according to the mentioned dynamic parameters may be valid for certain periods (especially in real environments).

This is controlled by the network round parameter. Because, if the best-chosen route is used continuously, the energy expenditure of the nodes and traffic intensity on that route can be inefficient. So, the proposed algorithm will always be finding the best solutions for certain rounds, and finding a global optimum cannot be a correct solution in these networks. In the classical ACO algorithm, the ants made the path selections according to Equation 3.16. However, this Equation was modified in accordance with the WSN structure in this paper and is introduced in Equations 3.17, 3.18 and 3.19.

$$P_{i,j}^{k}(t) = \frac{[\tau_{i,j}(t)]^{\alpha}[\varphi_{i,j}(t)]^{\beta}}{\sum_{k \epsilon allowed_k}[\tau_{i,k}(t)]^{\alpha}[\varphi_{i,k}(t)]^{\beta}} \; ; j \epsilon allowed_k \tag{3.16}$$

This method is started with transferring packets (ants) from an *S* node toward a destination node. As have been already mentioned, we may have multiple sources and destination nodes. Therefore, the algorithm can work in series, parallel and concurrent and find different efficient routes for each source and destination. Each ant is moving in accordance with the heuristic and pheromone rate values in choosing the path in each step. However, Equation 3.16 is revised as follows due to the dynamic nature of the WSNs. Thanks to this; the algorithm can be used in real-time applications. Therefore, Equations 3.17, 3.18 and 3.19, calculates the paths election by ants'. The search spaces are given in the deployment phase of the network.

If $E_j/E_{initial}$ >=60% and $Traffic_{i,j}$/Valid-Traffic >=50% and $Buffer_j$ is valid

$$P_{i,j}^{k}(t) = \frac{[\tau_{i,j}(t)]^{2\alpha}[\varphi_{i,j}(t)]^{\beta}}{\sum_{k \epsilon allowed_k}[\tau_{i,k}(t)]^{\alpha}[\varphi_{i,k}(t)]^{\beta}} \tag{3.17}$$

If 30<=$E_j/E_{initial}$<=60% and 20<=$Traffic_{i,j}$/Valid-Traffic <=50% and $Buffer_j$ is valid

$$P_{i,j}^{k}(t) = \frac{[\tau_{i,j}(t)]^{\alpha}[\varphi_{i,j}(t)]^{\beta}}{\sum_{k \epsilon allowed_k}[\tau_{i,k}(t)]^{\alpha}[\varphi_{i,k}(t)]^{\beta}} \tag{3.18}$$

Else

$$P_{i,j}^{k}(t) = \frac{[\tau_{i,j}(t)]^{\alpha}[\varphi_{i,j}(t)]^{3\beta}}{\sum_{k \epsilon allowed_k}[\tau_{i,k}(t)]^{\alpha}[\varphi_{i,k}(t)]^{\beta}} \tag{3.19}$$

Each ant performs its path according to the cost of the heuristic function and the pheromone ratio. Equations 3.20 and 3.21, respectively calculate them. At the end of each iteration, the algorithm avoids the local optimal problem by the evaporation technique based on Equation 3.22. This technique is used at the end of each iteration.

$$\varphi_{\mathrm{i,j}} = \mathrm{Cost}_{i,j} = \left(c_1 d_{i,j}\right) + \left(c_2 d_{j,BS}\right) + \left(c_3 \frac{\text{Valid−Traffic}}{\text{Traffic}_{i,j}}\right) + \left(c_4 \frac{E_{initial}}{E_j} . \frac{\text{Buffer Capisity}}{\text{Buffer}_{\mathrm{j}}}\right) \tag{3.20}$$

$$; c_1 < c_2 < c_3 < c_4 \;\; and \; c_1 + c_2 + c_3 + c_4 = 1$$

$$\tau_{\mathrm{i,j}} = \rho\tau_{\mathrm{i,j}} + \tau_{\mathrm{i,j}} \tag{3.21}$$

$$\tau_{\mathrm{i,j}} = \frac{1}{\tau_{\mathrm{i,j}}} \tag{3.22}$$

Where, c1, c2, c3, and c4 are coefficients. They are intended to calculate the cost by paying attention to the dynamic characteristics of the peer-to-peer connections between the nodes in the network with the importance level. The relationship between coefficients can be changed by the designer according to the requirement of applications. ρ is a coefficient that affects and controls the rate of pheromone released on the edges. α and β are related importance of trail and heuristic. $\tau_{\mathrm{i,j}}$ is the intensity of trails on the edges *i* and *j*. $\varphi_{\mathrm{i,j}}$ is the heuristic cost between the edges *i* and *j*. There is usually a trade-off between these two functions. In the heuristic function, $d_{i,j}$ presents the distance between *i* and *j* nodes, $d_{j,BS}$ presents the distance between *j* and base station that it is based straight line distance. $E_{initial}$ is the initial energy of each node that is supposed are the same in the first. E_j shows the current energy of the receiver node. $Traffic_{i,j}$ shows the current traffic intensity on the related edge between *i* and *j* nodes. The rate is important that is calculated based on valid traffic on each edge. The same concept and calculation are made according to the buffer size of the receiver node. Accordingly, it is learned how much buffer is available.

In the proposed algorithm, each ant goes from the source point to the destination node (e.g. base station) and returns again with the same route (Since ant holds a list of nodes that it chooses along the path, the return path can be selected correctly). In this case, each ant will be considered a tour. When all of the ants have completed a tour, one iteration will be completed. The number of ants (packets) and iterations will be defined in simulation. In real applications, especially, in distributed areas, they may not be clear at first. In each iteration, the same number of ants between S and D will be found in the solution (route). The algorithm will present the best of these as the solution to each iteration. The operating mechanism of the iterations is presented in Figure 3.49 as an example of the best route selection between each of the S and D nodes.

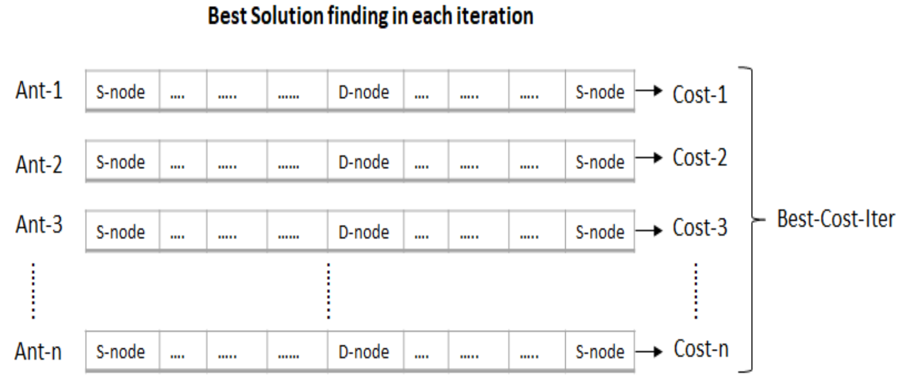


Figure 3. 48 At the end of each iteration the best solution is chosen based on ACO

3.5.2.2. Definition Packet Frame

The data frame template is refined that is shown in Figure 3.50. The Header section has been assigned to ant information. By identifying this section, the ants provide information on the path and use it in their route elections. In addition, the nodes in the routes will be able to update their data on their routing tables.

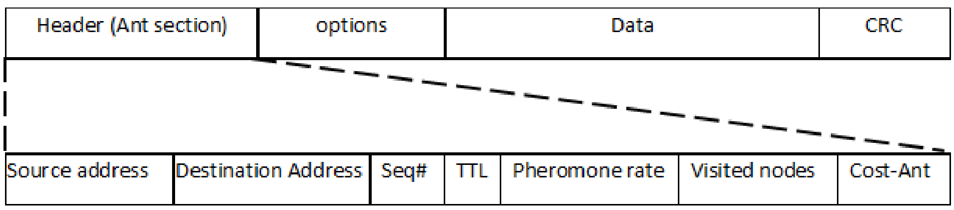


Figure 3. 49 Header section assigned to each ant

Seq# field is to control duplicate packets. Of course, this field can be ignored in general or in some circumstances depending on the definition of the problem (see Equations 9 and 10). The TTL field is intended to prevent the occupation of network traffic. Each node will decrease one unit per packet. The initial value of this field varies depending on the type of application. Thanks to the source and destination addresses filed, each ant can be applied to multiple sources and destination scenarios at the same time as parallel and concurrent because it knows which nodes are source and destination.

Therefore, ants from different S and D's do not interfere with each other. Pheromone rate is total pheromones that hold the entire path for each ant. Visited nodes field is a list of which nodes are passed via each ant. Cost-Ant filed shows the total cost of paths that are selected by each ant. It will be used for evaluation of the total route cost of each ant per tour (as mentioned in Figure 3.50).

Figure 3.51 shows how one tour ends with a simple example. When a tour is completed, the visited nodes by each ant and total route cost are calculated. Equation 3.23 obtains the Cost-Antk.

$$CostAnt_k = \sum_{edge=1}^{n} selectedcost_{edge}^{k} \tag{3.23}$$

Where n is the total number of elected edges in the path. The $selectedcost_{edge}^{k}$ is the cost of each elected edge by ants were gained from Equation 3.17, 3.18 or 3.19.

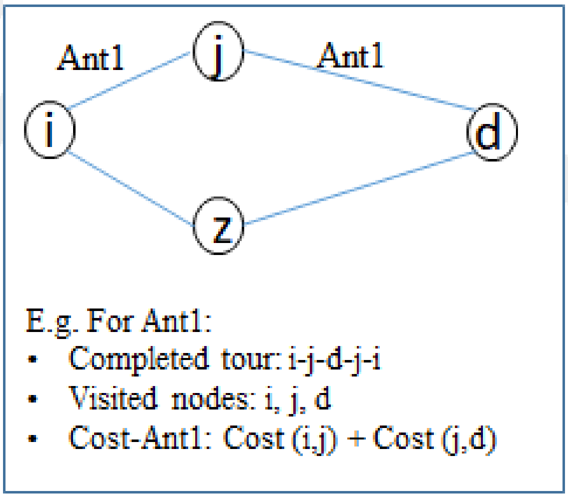


Figure 3. 50 An example of result of one tour for each ant

3.5.2.3. Definition Routing Table

The routing table of each node also refined which is shown in Figure 3.52. With this method, each node only holds its own neighbors' list, the pheromone ratio and the number of ants. In this protocol, the number of ants is not used in simulations. However, it is defined so that it can be used in required applications (e.g. for some analysis). Instead of creating two matrixes with node numbers dimensional for pheromones and heuristic costs, each node will hold only the information of its neighbors. It should be noted that the cost of the heuristic function varies in these networks (Unlike other applications). In each iteration, the nodes corresponding to the related paths, depending on the number of pheromones remaining, update the pheromone ratios and other required information in their routing tables.

Therefore, the memory and energy parameters of each node will be used efficiently and the life of the network will be extended. In addition, each ant will be able to retrieve the current pheromone and other information from the node where it resides. The issue that should not be forgotten that the dimensional size of each ant (the number of nodes between *S-D*) in WSNs may not be equal with each other.

Routing Table of node-i

Neighbor Lists	$d_{i,j}$	d_{BS}	Hop count$_{BS}$	Pheromone rate	Ant count
i	----	15	4	----	----
j	5	12	3	0.008	0
z	3	9	3	0.006	1

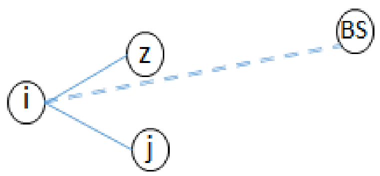


Figure 3. 51 Routing table format of nodes on the network. Node i is shown as an example

3.5.2.4. Other Features

Another feature of the proposed method is how to manage repeated packets. Depending on the application, it can discard these packets, or find a better route by checking the previous path of receiving that packet. These aims are able realized by Equations 3.24 and 3.25, respectively. Accordingly, each node will discard the packets when it receives them under one of some conditions as defined in Equation 3.24.

$$d_{\mathrm{i,BS}} \leq d_{j,BS}\ OR\ Hop_{\mathrm{i}} < Hop_{\mathrm{j}} \tag{3.24}$$

$$(CostAnt_k + d_{\mathrm{i,j}}) \leq d_{\mathrm{i,z}} \tag{3.25}$$

When node *i* receives a packet from node *j*, it checks whether it has been received before. If the discard mechanism is the application priority, it behaviors based on Equation 3.23. In this case, if the distance or hop-size of the node *j* to sink/BS is more than its distance, it discards the related packet. These methods are used in many types of the wireless networks and it is necessary. However, if they do not immediately discard the packet they received before (do not behavior greedy), they may find it more cost-effective to find other paths (as is defined in Equation 3.25).

Another feature is pheromone controls to balance resource consumption. If an edge (or a series of edges in a path) is used repeatedly as the best route, the resources consumed by the nodes on that path will be ineffective. To prevent this problem, we define a maximum value for pheromones on each edge. In this case, when the number of pheromones reaches to maximum on an edge, it overflows. In other words, the evaporation method in each iteration can be used as needed and thanks to it, the resources of the network can be used efficiently.

3.5.2.5. Definition Network Rounds

In order to apply ant colony-based methods to the actual applications of sensor networks, we must have unique considerations with the structure of these networks. In this protocol, we gave importance to this issue in all the steps and phases that developed. One of these features is to take into account the network round parameter. All the methods and features presented in this method can be used in various applications and can be used in monolithic areas. By identifying network rounds in certain times, we can reach various targets such as control of iterations, prevention of overload on the selected routes, updating of pheromones and so on. At the same time, the proposed algorithm can be used not only in simulation environments but also in real areas. The pseudo-code of the proposed algorithm is presented in Figure 3.53.

Algorithm(Parallel/ Concurrent/ Sequential)
While network-round is true While $Round_t$ <= $certain_{time}$ *Paths Election by Ants ;Eq.(3.17, 3.18, 3.19)* *Pheromone Update by Ant; Eq. (3.21)* *Routing Tables Update* *Calculate each path with its cost; Eq.(3.23)* *$Round_t$++* End $Round_t$=0 *Pheromone update based on evaporation ; Eq.(3.22)* End

Figure 3. 52 Pseudocode of proposed algorithm according to iteration numbers of simulation and network rounds

3.5.3. Simulation and Configuration Parameters

This section is to explain the simulation parameters of the proposed method. This method is also simulated in MATLAB. The input parameters are outlined in table 3.17. The input parameters are the same for all the algorithms in this section. The proposed method is compared with the ACO router chip (Okdem et. al., 2009), ACOHCM (Jiang & Zheng, 2018), LTAWSN (Mohajerani & Gharavian, 2016), Improved ACO approach (El Ghazi et. al., 2014).

We assume the time to be discrete t= (1, 2, …) and that at each time step each ant moves toward a neighbor node at a constant speed of one unit of length per time unit. The search spaces are given in the deployment phase of the network. Pheromone rate is a positive number in the initial (t=0) near to zero. Population and heuristic costs for all nodes and edges in the first iteration are gained based on the network deployment phase. Ant numbers are supposed 50, iteration numbers are also supposed 300 and it is run on 10 times. This is the same for all algorithms. As in the metaheuristic algorithm, there are some random parameters. In this case, we ran each algorithm 10 times and shows the result in average value.

Sensor nodes are deployed randomly, the number of sensor nodes is 100. The initial energy of the node is 0.1 J, the pheromone evaporation rate is 0. 5, Initial pheromone value is near to zero 0. 008. Energy consumption is supposed to receive and transfer that it is 2nj for each packet. Each packet size is 250 Kbyte and is transmitted in 600 microseconds. Network area is100×100. In this case, our upper band is [100, 100] and the lower band is [0, 0]. α=0.5, β=0.5, c1=0.15, c2=0.20, c3=0.25 and c4=0.4, ρ=0.5, $\tau_{i,j}(0)$=0.008. Valid traffic on each edge is 1. The maximum buffer size for each node is 10000 kB. The distance between each node and base station is calculated by RSSI or maximum likelihood methods. This algorithm is implemented on KIANI nodes (Kiani & Seyyedabbasi, 2018).

Table 3.17 Input parameters of each algorithm

Parameters	Values
Network Size	100*100 m
Number of ants	50
Initial energy	0.1 J
Packet size	250byte
Energy consumption in receive and transfer	2 nj
Initial pheromone value	0.008
Pheromone evaporation rate ρ	0.5
α	0.5
β	0.5

Also, the sensor nodes are static and deployed randomly. The sink/BS has unlimited energy supply and located in the center of the network. In the network, all the sensors have static energy in the initialization. All the sensors are homogenous and have the same energy level and transmission range. These network assumptions and input parameters are the same for all algorithms in the simulation. Just compared algorithms have some parameters that special them. So, they are different. Evaluation of algorithm based on metric parameters such as; the number of alive nodes, average energy consumption.

3.5.4. Comparison and Discussion

In this section, the metric performance of the proposed method is evaluated. As mentioned in the previous section, the input parameters are the same for all algorithms. The proposed metaheuristic algorithm is a modified version of the ACO algorithm. The function probability for each ant to choose the next-hop has a difference with the main ACO algorithm. Also, the pheromone evaporation is modified. Furthermore, the proposed method has a specific routing scheme in data packet transmission.

Figure 3.54 shows the network lifetime of different algorithms improved by ACO in WSN routing algorithms. Figure 3.54 highlights five routing algorithms network lifetime such as; ACO approach, Improved ACO approach, LTAWSN, ACOHCM, and our proposed method. Here these algorithms run with different sensor node numbers. Respectively from 50, 60, 70, 80, 90 and 100.

In this figure, we illustrated the first dead sensor node in the whole network lifetime. As shown in Figure 3.54, in our proposed method the first dead sensor node when the network has 50 sensor nodes is in 53 iterations of the network. Also when the network has 100 sensor nodes, the first dead sensor node is in 260 iterations of a network. Table 3.18 lists the data of each iteration and first dead sensor nodes.

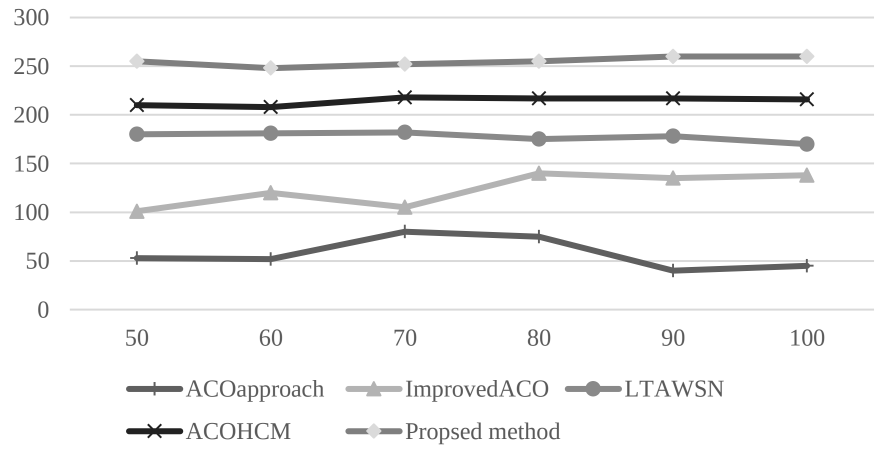


Figure 3. 53 The network lifetime in different iteration versus sensor node size

Table 3.18 Iteration and first dead sensor nodes

Sensor	ACOapproach	Improved ACO	LTA WSN	ACOHCM	Proposed method
50	53	101	180	210	255
60	52	120	181	208	248
70	80	105	182	218	252
80	75	141	175	217	255
90	40	135	178	217	260
100	45	138	170	216	260

As energy consumption is the main factor of each routing algorithm's functionality, Figure 3.55 shows the proposed algorithm and the other four algorithms. Here the convergence speed of each algorithm illustrated. A routing algorithm has good functionality and draws low energy that has a high convergence speed, so this algorithm has a high success rate of searching for the best rout path. Figure 3.56 shows a clear trend in convergence speed. Our proposed method finds the best path after 55 iterations on average of 10 times run.

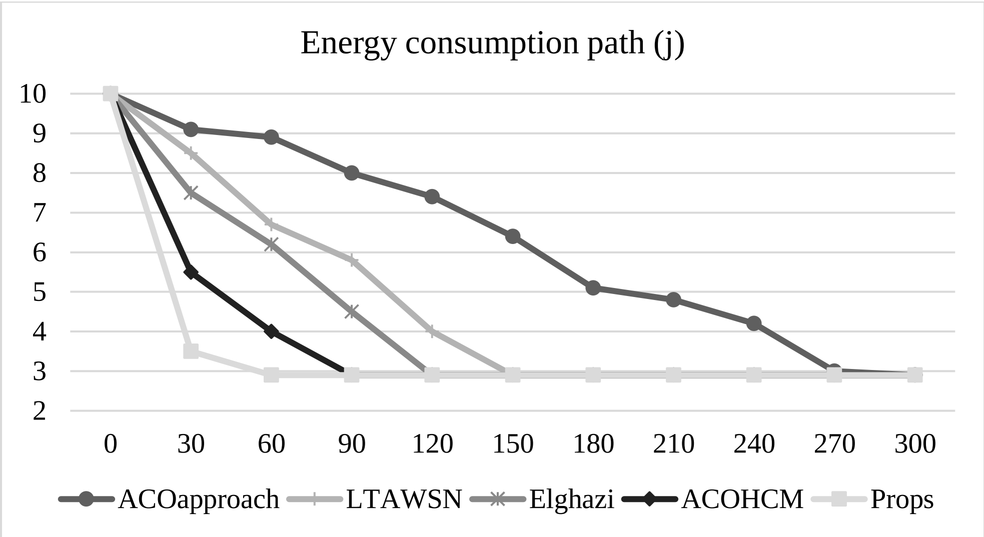


Figure 3. 54 Convergence graph of energy consumption in iteration versus on total energy of whole network

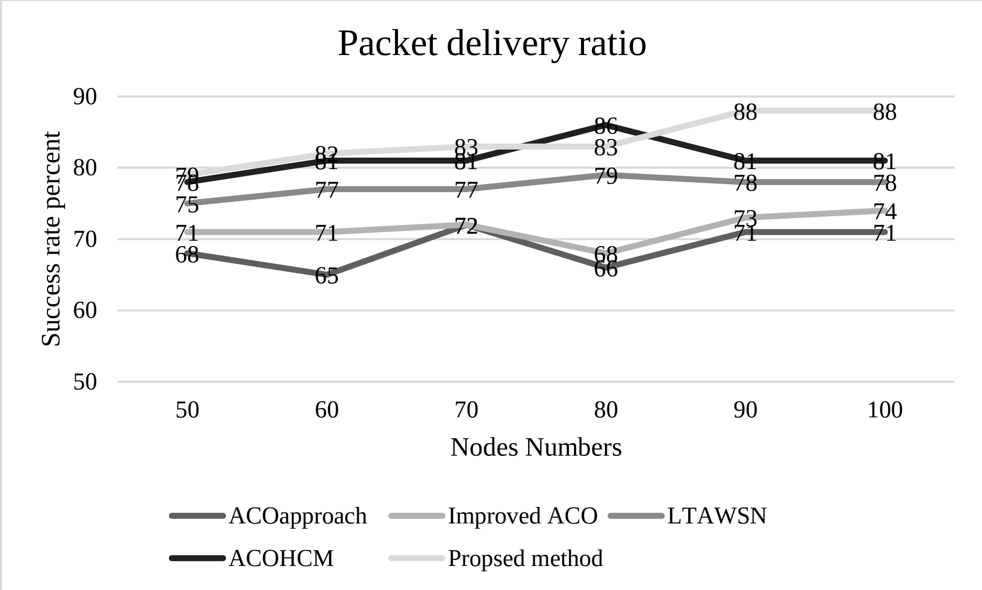


Figure 3. 55 Packet delivery ratio in different algorithms

Figure 3.56 gives a brief report about the packet delivery in the proposed method. These results obtained from different sensor numbers in the same network parameters. The proposed method has good performance in comparison with others. Due to the routing scheme in the proposed method. In this method, the mechanism control each of the packets delivery also reduce the data packets redundancy. Obviously, reducing duplicate data packets save more energy in each sensor node. As a result, the network lifetime is increased. Besides, the proposed method has a different probability function to choose the next hop. This function takes into important parameters such as the energy of the sensor node and traffic of the path.

3.5.5. Conclusion

Energy consumption is one of the main problems in routing protocols of wireless sensor networks (WSN). Sensor nodes have limited battery level and memory space, it is important to manage these resources in an optimal way. The resources in wireless have an important role in network efficiency. The network lifetime has a direct relation with sensor nodes' residual energy. One important metric factor is data packet delivery. The network should reduce the lost data packets. Since each of these packets have important data collected from the area. Consider an event-based network that should transfer global knowledge of an event to the sink/BS. So, the network connectivity is important, for this reason, connectivity must always be established to ensure that information is delivered to the destination.

This method is a new routing protocol based on ant colony optimization that manages efficient network resources. Our proposed method has been modified to find the next destination of ants also pheromone update and evaporation rate operators. We consider other parameters in choosing the next destination in different conditions of a destination like residual energy, buffer size, traffic rate, and distance. The proposed method finds the optimal path with low energy consumption also by prolonging the network lifetime.

The main goal of metaheuristic algorithms is to find the best solutions. Our proposed method based on the metaheuristic algorithm ACO. The ACO based on ants' behavior in nature. The population is important in the metaheuristic algorithms. In the ACO algorithms ants consisted of the population of the algorithm. The ant search in the solutions to find the best and optimal solution based on protocol goal. The maximization or minimization are the goals of protocols. If the best solutions are the maximum value, the ants try to find the best maximum solution, the contrary tries to find the minimum optimal solution. Most of the problems aim to minimize as in WSN applications. In the proposed method ants tries to find the minimum cost path in terms of consuming little energy, choose the path with low traffic and a small distance. The pheromone that each ant release in the path can be useable in finding an optimal path after some iteration. Our proposed method finds the best optimal path by ants helps us to transfer data packets with low cost.

The routing method here follows three conditions, in other meaning; ants should choose the path from the destination situation. Traffic rate, energy level, distance, and buffer size are the parameters have an effect on pathfinding. As mentioned before, the probability function to choose next-hop is revised in this method. The heuristic and pheromone values have an impact on this function too. The main problem of the metaheuristic algorithms is a trapping on local optima. ACO avoids this problem by evaporation mechanism. As each ant leaves a level of pheromone along the path. This is necessary to update this value by a mechanism. The pheromone ratio and evaporation follow equation that explained in the algorithm description section. The heuristic function is the cost function of the algorithm. In the proposed method, heuristic function calculated from these parameters to give a value to the ant that wants move from node *i* to *j*. Distance between nodes *i* and *j*, the distance between node *j* to the sink/BS, the traffic ratio between two nodes and the energy level and buffer size of the node *j*. In addition, there are four coefficients between 0 and 1 for these parameters. In the routing, each ant has the best solution for the destination in the iteration. If ant start from node *D* and the source is the node *S*. After one iteration is completed, there are *n* path solution from node *D* to *S* (n is the number of ants). Then, the ACO algorithm chose the minimum cost path to send data packet from two nodes *D* and *S*.

The data packet frame contains the path nodes. In the data packet frame, there are information about the source and destination address, time to live, pheromone rate and visited ants. Each sensor nodes have a routing table that keeps information about ants. As the memory of the sensor nodes is limited, they just keep the neighbor nodes' information. In the routing, each ant updates the path by retrieving routing tables. Routing tables contain neighbors’ list, distances between neighbors and the sink/BS, hop count to sink/BS, pheromone rate and ant count that visited the sensor node. The visited ants are important that each ant can update its own route by accessing the sensor nodes routing table.

Also, the proposed method increases the data packet delivery and reduce the redundancy. Consider node *i* send a data packet to node *j*. the node j checks the data packet. If it received before the discard mechanism applies. If the distance or hop-size of the node *j* to sink/BS is more than its distance, the data packet discards. Otherwise, the discard mechanism can find another path. Also, there is the maximum value of the pheromone on each edge to prevent repeatedly using the special path. Although there is an evaporation rate in the ACO. The maximum pheromone value causes consume the resources efficiently.

The proposed method in comparison with other routing algorithms based on ACO has well results in network lifetime and energy consumption. In the simulation, all of the input parameters for algorithms are the same. The simulation results prove that the proposed method has good performance in energy efficiency. Also, the main goal of this algorithm to use resources efficiently. Besides the packet delivery ratio is noticeable. The proposed method network has a long lifetime since the ants chose the nodes based on the destination energy level. The energy level is not one parameter in destination choosing. The distance also has an effect. As all sensors run in low communication range. In this way, the distance is important too. The proposed method results prove that this algorithm can have good results in concurrent and real-time data transfer on a big scale of wireless sensor networks. The proposed algorithm will perform better on larger-scale networks. Since connectivity is higher and longer than other algorithms. In simulation results, from 300 nodes, the first node dies in 260_{th} round of the network.

CHAPTER FOUR: CONCLUSIONS AND FUTURE WORKS

Many of many small sensor nodes consisted of a wireless sensor network (WSN). The sensor node works cooperatively based on functionality to collect data from an area. Usually, they used to track or monitor an event. The events are different in applications. The sensor nodes have restrictions on the sensor nodes like; limited energy and memory. Each sensor node have capability to sensing, computation and communication. The energy consuming in the wireless sensor networks is the main challenges. There are different approaches to overcome in this issue. Routing algorithms, data aggregation, clustering schemes are the difference approaches. Network lifetime is important to enhance connectivity, scalability and reliability of the network. This thesis focused on two schemes of energy efficiency on WSN; routing algorithms and cluster schemes.

There are different applications of WSN that designed for specific functionality. Most of the applications established in harsh environments since, sensor nodes technology is easy to use in harsh environments, low cost and easy to deploy. The main disadvantages are the unchangeable batteries. Also, the challenge in WSN is energy-consuming. In this way, the routing algorithms and clustering schemes are proposed. The main purpose of the thesis to reduce energy consumption. As a result, the network lifetime, packet delivery, scalability are increasing. Besides, in the thesis two metaheuristic algorithms are proposed I-GWO and Ex-GWO that inspired by the GWO algorithm. The clustering and routing are performed with these algorithms, besides the modified version of the ACO algorithm are proposed to use network resources efficient in routing methods. Briefly, the thesis is proposed five methods that proposed two clustering methods and three routing algorithms. One of the routing algorithms is proposed to heterogeneous wireless sensor networks. Other algorithms are homogeneous sensor nodes.

The routing techniques are divided into three; flat-based, hierarchical-based, and location-based protocols. In the flat-based topologies, all of the sensors have the same role. Each sensor can communicate with the base station. The hierarchical-based topologies follow the cluster schemes. In this scheme, there are clusters and cluster heads. Each cluster has at least one sensor node. The cluster members transfer data packets to the cluster heads. They are responsible for communication with the base station. In the location-based protocols each sensor equipped with the GPS module to be aware of its own location in the network. This type of network is not energy efficient protocols since sensor nodes consume more energy in location findings. Globally, most of the routing protocols based on these three topologies.

As the main goal of the thesis is energy efficiency in the WSN. The clustering methods also discussed. The cluster methods main aim to reduce energy consuming. There is a fitness function to choose the best and optimal cluster head from candidate sensor nodes. In most of the cluster head selection methods, the authors supposed the residual energy as a parameter in the selection phase. In our proposed methods, we consider other parameters in the cluster head selection phase. Most clustering methods follow the TDMA schedule in the data routing between sensor nodes. TDMA schedule avoids data collisions in the network. The cluster members transfer data packets in the specific time intervals to the cluster heads. Also, cluster heads follow this schedule too. They collect data from cluster members, then aggregate data packet. Finally, the aggregated data is transfer in time intervals to the base station. The goal of clustering methods is reducing the communication number with data aggregation.

In the first proposed method, we suggested a new clustering method. The network topology in the HEEL is hierarchical-based. The clustering scheme is popular in this type of topology. The main goal of clustering methods is to enhance the network's lifetime. The clustering method reduces the communication of the sensor nodes with the base station by cluster head. Cluster head only responsible for the data packet transfer. The most clustering algorithms are variant versions of the LEACH algorithm. In the HEEL, there are four parameters to select optimal cluster head. There are static clusters. Each cluster has at least one sensor node. In the cluster head selection phase, four parameters such as; residual energy, the distance between a sensor node and the base station and the number of links to neighbors have an impact in cluster head selection. Each of these parameters has a static coefficient. Each coefficient has an impact on the parameters value. In the HEEL, the fitness function, select the best cluster head from all candidate sensor nodes. We consider the entire sensor nodes in each round of networks can be a cluster head. In each round of the HEEL, the network has static clusters and cluster heads. The obtained results from the simulation determine that the HEEL algorithm provides energy efficiency in the WSN. The proposed method can be used in events monitoring.

The second proposed method is an efficient routing algorithm in heterogeneous wireless sensor networks (HWSN). EEHRSN is based on the hierarchical based topology. There are three different types of sensor nodes. Normal, advanced, and super sensor nodes. The heterogeneity of these sensors in the battery level, communication range, and memory space. The EEHRSN consists of the step, the clustering, and routing. In the clustering, the network is divided into four regions, and then each region is divided into four clusters again. Therefore, the network has sixteen virtual clusters. In the center of each region, four super node is located. In the center of each sixteen cluster, the same number of clusters the advanced nodes located. The advanced and super nodes act as cluster head and region head respectively. Totally, there are twenty cluster heads in the network. EEHRSN is an event-based protocol. When, the normal nodes detect an event, transfer event information to the cluster heads. Then cluster heads transfer the information via super nodes to the base station. In the routing phase, only the cluster heads (advanced nodes) and region heads (super nodes) communicate with the base station. The cluster and region heads have more energy than normal nodes. In the network initialization, the normal and advanced nodes have an identical number of appropriate cluster heads and region heads respectively. There are three alternative paths from advanced nodes to super nodes. The advanced node can transfer the data packet to the sink/BS directly when there are not available super nodes. Also, the super nodes have four alternative paths to transfer data packets. They can benefit from the advanced node to data transmission. The simulation results obtained that, the EEHRSN algorithm has an improvement in data packet delivery, network lifetime. In the network lifetime, the EEHRSN has noticeable performance since until last iterations there are alive sensor nodes. As a result, the packet delivered to the base station is increased too. The EEHRSN algorithm is a heterogeneous network. As most IoT applications have heterogeneous things, the proposed method is a good choice for IoT applications.

The third method is an improved version of the HEEL algorithm (I-HEEL) by metaheuristic algorithms. I-HEEL is a clustering method that applied with three metaheuristic algorithms GWO, I-GWO and Ex-GWO. The metaheuristic algorithms proposed by the author of the thesis. GWO algorithm is inspired by the behaviors of the grey wolves in nature. The algorithm simulated the encircling, hunting and attacking habitats of wolves. I-GWO and Ex-GWO are modified versions of the GWO in the encircling phase. I-GWO is based on the first wolf in the pack as a leader; remaining wolves in the pack update its position based on the alpha wolf position. The algorithm considers that the alpha position is located in the best position of the hunt. Incremental grey wolf optimization (I-GWO) has the best performance than the GWO algorithm in the encircling phase. In the Ex-GWO algorithm, the nth benefits the n-1 wolf before it to update its position. In the expanded grey wolf optimization (Ex-GWO) the main goal is to benefit the swarm intelligent in position updating. In the HEEL algorithm, there are four static coefficients. I-HEEL tries to update these coefficients by the GWO, I-GWO and Ex-GWO. So, the coefficients update dynamically in each round of the network. Furthermore, GWO, I-GWO and Ex-GWO algorithms also perform the cluster head selection. I-HEEL has a better performance than the HEEL algorithm. Since the static coefficients in the HEEL are disadvantage of this algorithm. I-HEEL based on the Ex-GWO algorithm can extend the network lifetime until the 451st round of 500 rounds. I-HEEL based on the I-GWO and Ex-GWO has the best performance, but the Ex-GWO structure has the main effect on the efficiency of the I-HEEL. I-HEEL based on the I-GWO and Ex-GWO evaluated in the metrics such as; network lifetime, residual energy and data packet ratio. In both of them, there are noticeable enhance in comparison of HEEL. Although the I-GWO execution time is faster than the I-GWO algorithm, the Ex-GWO is successful in the clustering method.

The fourth method of the thesis proposed a new routing algorithm by metaheuristic algorithms, I-GWO and Ex-GWO. The network topology is flat-based topology. The routing is this algorithm is multi-hop. The proposed method considers the energy, traffic, buffer, and distance, and hop size to find the best path between destination and base station. In the flat-based network topologies, all sensors have the same and similar roles. There are not any cluster head in this network. As the main challenge in the WSN is energy consumption. In this method, we focused on this to solve by routing algorithms. The metaheuristic algorithms have good performance in finding the best solutions. They do not guarantee to find the best solution by tries to find the solution to avoid the trap on the local optima. There is a fitness function to decide which path is feasible to use. The operation to find optimal path is performed in the base station, since it has unlimited energy supply and high computation capacity. In the best optimal pathfinding with two nodes, the residual energy of destination node, the distance between two nodes, hop size to the base station and of the destination node traffic status between two nodes is important. There is two information table in the network, the routing table is stored in each node memory and sensor nodes global information keeps on the base station. The base station with this information decides to choose the minimum cost path. Then broadcast the path to the source nodes. There are different data packets to transfer to useful information between node and base station. According to the obtained result, the routing algorithm based on the I-GWO and Ex-GWO has good performance in comparison to the routing algorithm inspired by the GA, BEE and ACO algorithms. The routing algorithm efficiency is evaluated by metrics such as; the network lifetime, successful data packet delivery, number of lost data packets. In this metric, the Ex-GWO has better than the I-GWO algorithm.

The last method is concurrent pathfinding in real-time wireless sensor networks by a new routing protocol inspired by ant colony optimization. The network is flat-based topology. All of the sensor nodes have fixed and the same initial energy level. The goal in this algorithm to achieve a balance by using resources of the network. In this algorithm, there is a modified version of the ACO algorithm. In the ACO algorithm, the pheromone and heuristic value is important on the probability function to choose the next hop. The function probability in this version of the ACO is also modified and ants decide to choose the next hop in three situations. The situation of the destination sensor node. Ants in the colony find the path with pheromone value in each edge; the heuristic value is defined also. In the proposed method, all ants that completed a tour, the best path with minimum cost value is selected as the best path. The ant decides to choose the next hop with four parameters such as; residual energy, buffer size, traffic rate, and distance. The minimum path is the path with the minimum energy consuming. The main goal of the algorithm to use the resource in an optimal way. The previously defined data packets in the network are responsible for the data packet transferring. The proposed method is successful in fault tolerance. There is a solution to handle the data packet on the unavailable sensor nodes. Although it has a high cost, the packet delivery ratio is increased. After all, the proposed method in comparison to the routing algorithm with ACO have good performance. The first dead node is in the 250th round of the network round. The network round in the simulation is 300.

As mentioned, there are two clustering method and there routing algorithm are proposed in thesis. Here, there are brief comparison with these algorithms. The HEEL and I-HEEL are two clustering methods. HEEL in comparison of the I-HEEL is consume more energy. Also, the network life in the I-HEEL is longer than the HEEL algorithm. Since I-HEEL applied with two metaheuristics. The metaheuristic algorithms tries to find the best solution in the search space. I-HEEL based on the I-GWO and Ex-GWO algorithm update coefficients dynamically based on the network status. As a result, the metaheuristic algorithm has efficiency in the clustering methods.

Also, the routing algorithm described in the thesis is; the heterogeneous routing methods (EEHRSN), routing algorithm based on the I-GWO and Ex-GWO algorithm, also the routing algorithm based on the modified ACO algorithm. The network lifetime in the EEHRSN is longer than two other algorithms. Since in the EEHRSN there are different sensor types with various initial energy. Consequently, the successful data packet delivery is higher than others. Two routing algorithms by metaheuristic algorithms are evaluated too. Although both of them have different metaheuristic structure. The routing algorithm based on the modified ACO has a different convergence curve in energy consumption. This algorithm, finds the best path after 20% percent of the network rounds, due to the pheromone rate on each edge. While I-GWO and Ex-GWO find the path in each round separately. However, the routing algorithm based on the Ex-GWO has good performance than others. This is the result of the structure of the Ex-GWO. Since they used all wolves' knowledge in the pack. The modified ACO has good performance in network stability. In means of energy consumption, the network after 20% round run on determined energy until all sensors dead. Finally, the packet delivery ratio in the routing algorithm based on the Ex-GWO is higher than other algorithms. Since the routing method has strong data packet routings. The global information of each sensor node has an impact on data routing. Also, the main point is that the routing in the I-GWO and Ex-GWO has a maximum routing hop. This number is determined by the maximum hop size of the sensor in the network.

As future works;

- Improve the TEEN algorithm based on the EEHRSN algorithms, in terms of three-level heterogeneity
- Use machine learning method such as Q-learning, reinforcement learning, ANN to updates coefficients on the HEEL
- Hybrid I-GWO and Ex-GWO with modified ACO to propose anew routing algorithm
- Declare the cluster number with I-GWO and Ex-GWO algorithms
- Extend the I-HEEL protocol in the mobile sensor networks

REFERENCES

A. Manjeshwar and D. P. Agrawal. Apteen: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In Proceedings of the 16th International Parallel and Distributed Processing Symposium (IPDPS), page 48, Washington, DC, USA, 2002b. IEEE Computer Society.

A. Manjeshwar and D. P. Agrawal. Teen: A routing protocol for enhanced efficiency in wireless sensor networks. In Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS), page 189, Washington, DC, USA, 2001a. IEEE Computer Society.

Akyildiz, I. F., & Vuran, M. C. (2010). Wireless sensor networks (Vol. 4). John Wiley & Sons.

Akyildiz, I. F., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. IEEE Communications Magazine, 40 (8), 102-114.

Al-Baz, A., & El-Sayed, A. (2018). A new algorithm for cluster head selection in LEACH protocol for wireless sensor networks. International journal of communication systems, 31(1), e3407.

Anastasi, G., Conti, M., Di Francesco, M., & Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. Ad hoc networks, 7(3), 537-568.

Ari, A. A. A., Yenke, B. O., Labraoui, N., Damakoa, I., & Gueroui, A. (2016). A power efficient cluster-based routing algorithm for wireless sensor networks: Honeybees swarm intelligence based approach. Journal of Network and Computer Applications, 69, 77-97.

Boukergche, A., Ahmad, M. Z., Turgut, B., & Turgut, D. (2008). A taxonomy of routing protocols in sensor networks. Algorithms and Protocols for Wireless Sensor Networks, 129-160.

Camilo, T., Carreto, C., Silva, J. S., & Boavida, F. (2006, September). An energy-efficient ant-based routing algorithm for wireless sensor networks. In International workshop on ant colony optimization and swarm intelligence (pp. 49-59). Springer, Berlin, Heidelberg.

Chawla, P., Chana, I., & Rana, A. (2015). A novel strategy for automatic test data generation using soft computing technique. Frontiers of Computer Science, 9(3), 346-363.

Chen, B., Jamieson, K., Balakrishnan, H., & Morris, R. (2002). Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. Wireless networks, 8(5), 481-494.

Chen, J., & Shen, H. (2008, October). MELEACH-L: More energy-efficient LEACH for large-scale WSNs. In 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing (pp. 1-4). IEEE.

Costa, D., & Hertz, A. (1997). Ants can colour graphs. Journal of the operational research society, 48(3), 295-305.

Deshpande, A., Guestrin, C., Madden, S. R., Hellerstein, J. M., & Hong, W. (2004, August). Model-driven data acquisition in sensor networks. In Proceedings of the Thirtieth international conference on Very large data bases-Volume 30 (pp. 588-599). VLDB Endowment.

Dogan, G. (2016). A Survey of Provenance in Wireless Sensor Networks. Adhoc & Sensor Wireless Networks, 31.

Dorigo, M., & Di Caro, G. (1999, July). Ant colony optimization: a new meta-heuristic. In Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406) (Vol. 2, pp. 1470-1477). IEEE.

Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science (pp. 39-43). Ieee.

El Ghazi, A., Ahiod, B., & Ouaarab, A. (2014, May). Improved ant colony optimization routing protocol for wireless sensor networks. In International Conference on Networked Systems (pp. 246-256). Springer, Cham.

Elhoseny, M., Elminir, H., Riad, A., & Yuan, X. (2016). A secure data routing schema for WSN using elliptic curve cryptography and homomorphic encryption. Journal of King Saud University-Computer and Information Sciences, 28(3), 262-275.

Gambardella, L. M., Taillard, É., & Agazzi, G. (1999). Macs-vrptw: A multiple colony system for vehicle routing problems with time windows. In New ideas in optimization.

Gautam, N., & Pyun, J. Y. (2010). Distance aware intelligent clustering protocol for wireless sensor networks. Journal of communications and networks, 12(2), 122-129.

Guha, S., Biswas, K., Ford, B., Sivakumar, S., & Srisuresh, P. (2008). NAT Behavioral requirements for TCP. RFC 5382 (Best Current Practice).

Gupta, S. K., Kuila, P., & Jana, P. K. (2013, February). GAR: an energy efficient GA-based routing for wireless sensor networks. In International conference on distributed computing and internet technology (pp. 267-277). Springer, Berlin, Heidelberg.

Hammoudeh, M., & Newman, R. (2015). Adaptive routing in wireless sensor networks: QoS optimisation for enhanced application performance. Information Fusion, 22, 3-15.

He, T., Stankovic, J. A., Lu, C., & Abdelzaher, T. (2003). SPEED: A stateless protocol for real-time communication in sensor networks. VIRGINIA UNIV CHARLOTTESVILLE DEPT OF COMPUTER SCIENCE.

Heinzelman, Wendi Rabiner, Anantha Chandrakasan, and Hari Balakrishnan. "Energy-efficient communication protocol for wireless microsensor networks." System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on. IEEE, 2000.

Holland, J. H. (1992). Genetic algorithms. Scientific american, 267(1), 66-73.

Intanagonwiwat, C., Govindan, R., & Estrin, D. (2000, August). Directed diffusion: A scalable and robust communication paradigm for sensor networks. In Proceedings of the 6th annual international conference on Mobile computing and networking (pp. 56-67). ACM.

Jamal, N. E., & Kamal, A. (2004). Routing techniques in wireless sensor networks: A survey. IEEE Wireless Communication, 11, 6-28.

Jiang, A., & Zheng, L. (2018). An effective hybrid routing algorithm in wsn: Ant colony optimization in combination with hop count minimization. Sensors, 18(4), 1020.

Joshi H, Arora S (2017) Enhanced grey wolf optimization algorithm for global optimization. Fundam Inform 153(3):235–264

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. Applied soft computing, 8(1), 687-697.

Khoshkangini, R., Zaboli, S., & Conti, M. (2014, September). Efficient routing protocol via ant colony optimization (ACO) and breadth first search (BFS). In Proceedings of the 2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom), Taipei, Taiwan (pp. 1-3).

Kiani, F. (2014). Designing New Routing Algorithms Optimized for Wireless Sensor Network. LAP LAMBERT Academic Publishing, Dusseldorf, Germany.

Kiani, F., & Seyyedabbasi, A. (2018). Wireless sensor network and internet of things in precision agriculture. Int. J. Adv. Comput. Sci. Appl, 9(8), 220-226.

Kiani, F., Amiri, E., Zamani, M., Khodadadi, T., & Abdul Manaf, A. (2015). Efficient intelligent energy routing protocol in wireless sensor networks. International Journal of Distributed Sensor Networks, 11(3), 618072.

Kiani, F., Rad, A., Sis, M. K., Kut, A., & Alpkocak, A. (2013). EEAR: an energy effective-accuracy routing algorithm for wireless sensor networks. Life Science Journal, 10(2), 39-45.

Kim, S., Pakzad, S., Culler, D., Demmel, J., Fenves, G., Glaser, S., & Turon, M. (2006, October). Wireless sensor networks for structural health monitoring. In Proceedings of the 4th international conference on Embedded networked sensor systems (pp. 427-428). ACM.

Kulik, J., Heinzelman, W., & Balakrishnan, H. (2002). Negotiation-based protocols for disseminating information in wireless sensor networks. Wireless networks, 8(2/3), 169-185.

Kulkarni, R. V., Forster, A., & Venayagamoorthy, G. K. (2010). Computational intelligence in wireless sensor networks: A survey. IEEE communications surveys & tutorials, 13(1), 68-96.

Kumar, D., Aseri, T. C., & Patel, R. B. (2009). EEHC: Energy efficient heterogeneous clustered scheme for wireless sensor networks. Computer Communications, 32(4), 662-667.

Kumar, D., Aseri, T. C., & Patel, R. B. (2011). Multi-hop communication routing (MCR) protocol for heterogeneous wireless sensor networks. International Journal of Information Technology, Communications and Convergence, 1(2), 130-145.

Lee, J. S., & Kao, T. Y. (2016). An improved three-layer low-energy adaptive clustering hierarchy for wireless sensor networks. IEEE Internet of Things Journal, 3(6), 951-958.

Lee, J. W., Choi, B. S., & Lee, J. J. (2011). Energy-efficient coverage of wireless sensor networks using ant colony optimization with three types of pheromones. IEEE Transactions on Industrial Informatics, 7(3), 419-427.

Li, L., & Halpern, J. Y. (2001, June). Minimum-energy mobile wireless networks revisited. In ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No. 01CH37240) (Vol. 1, pp. 278-283). IEEE.

Li, P., Nie, H., Qiu, L., & Wang, R. (2017). Energy optimization of ant colony algorithm in wireless sensor network. International Journal of Distributed Sensor Networks, 13(4), 1550147717704831.

Lin, C., Wu, G., Xia, F., Li, M., Yao, L., & Pei, Z. (2012). Energy efficient ant colony algorithms for data aggregation in wireless sensor networks. Journal of Computer and System Sciences, 78(6), 1686-1702.

Lindsey, S., & Raghavendra, C. S. (2002, March). PEGASIS: Power-efficient gathering in sensor information systems. In Proceedings, IEEE aerospace conference (Vol. 3, pp. 3-3). IEEE.

Liu, Y., Yang, Z., Ning, T., & Wu, H. (2014). Efficient quality-of-service (QoS) support in mobile opportunistic networks. IEEE Transactions on Vehicular Technology, 63(9), 4574-4584.

Loscri, V., Morabito, G., & Marano, S. (2005, September). A two-levels hierarchy for low-energy adaptive clustering hierarchy (TL-LEACH). In IEEE vehicular technology conference (Vol. 62, No. 3, p. 1809). IEEE; 1999.

Lv, Y. (2014). An energy efficient routing based on swarm intelligence for wireless sensor networks. Journal of Software, 9(10).

M. Ilyas and I. Mahgoub. Handbook of sensor networks: compact wireless and wired sensing systems. CRC press, USA, 2005.

Mahmood, D., Javaid, N., Mahmood, S., Qureshi, S., Memon, A. M., & Zaman, T. (2013, October). MODLEACH: a variant of LEACH for WSNs. In 2013 Eighth International Conference on Broadband and Wireless Computing, Communication and Applications (pp. 158-163). IEEE.

Maniezzo, V. (1999). Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. INFORMS journal on computing, 11(4), 358-369.

Maniezzo, V., & Colorni, A. (1999). The ant system applied to the quadratic assignment problem. IEEE Transactions on knowledge and data engineering, 11(5), 769-778.

Martorosyan, A., Boukerche, A., & Pazzi, R.W. (2008). A taxonomy of cluster-based routing protocols for wireless sensor networks. International Symposium on Parallel Architectures, Algorithms, and Networks. 3 (5), 247-253.

Mirjalili, S. (2015). The ant lion optimizer. Advances in Engineering Software, 83, 80-98.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software, 69, 46-61.

Mittal, N., Singh, U., & Sohi, B. S. (2016). Modified grey wolf optimizer for global engineering optimization. Applied Computational Intelligence and Soft Computing, 2016, 8.

Mohajerani, A., & Gharavian, D. (2016). An ant colony optimization based routing algorithm for extending network lifetime in wireless sensor networks. Wireless Networks, 22(8), 2637-2647.

Okdem, S., & Karaboga, D. (2009). Routing in wireless sensor networks using an ant colony optimization (ACO) router chip. Sensors, 9(2), 909-921.

Omar, D. M., & Khedr, A. M. (2018). ERPLBC-CS: Energy Efficient Routing Protocol for Load Balanced Clustering in Wireless Sensor Networks. Adhoc & Sensor Wireless Networks, 42.

Qing, L., Zhu, Q., & Wang, M. (2006). Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks. Computer communications, 29(12), 2230-2237.

Rajakumar, R., Amudhavel, J., Dhavachelvan, P., & Vengattaraman, T. (2017). GWO-LPWSN: Grey wolf optimization algorithm for node localization problem in wireless sensor networks. Journal of Computer Networks and Communications, 2017.

Rodoplu, V., & Meng, T. H. (1999). Minimum energy mobile wireless networks. IEEE Journal on selected areas in communications, 17(8), 1333-1344.

Rostami, A. S., Badkoobe, M., Mohanna, F., Hosseinabadi, A. A. R., & Sangaiah, A. K. (2018). Survey on clustering in heterogeneous and homogeneous wireless sensor networks. The Journal of Supercomputing, 74(1), 277-323.

Selvakennedy, S., Sinnappan, S., & Shang, Y. (2006). T-ANT: a nature-inspired data gathering protocol for wireless sensor networks. Journal of Communications, 1(2), 22-29.

Seyyedabbasi, A., & Kiani, F. (2019). I-GWO and Ex-GWO: improved algorithms of the Grey Wolf Optimizer to solve global optimization problems. Engineering with Computers, 1-24.

Seyyedabbasi, A., Koç, M., & Kiani, F. (2016). Information of road events by using wireless sensor networks to vehicles.

Smaragdakis, G., Matta, I., & Bestavros, A. (2004). SEP: A stable election protocol for clustered heterogeneous wireless sensor networks. Boston University Computer Science Department.

Sohraby, K., Minoli, D., & Znati, T. (2007). Wireless sensor networks: technology, protocols, and applications. John Wiley & Sons.

Stützle, T., & Dorigo, M. (1999). ACO algorithms for the traveling salesman problem. Evolutionary algorithms in engineering and computer science, 163-183.

Talbi, E. G. (2009). Metaheuristics: from design to implementation (Vol. 74). John Wiley & Sons.

Tanwar, S., Kumar, N., & Rodrigues, J. J. (2015). A systematic review on heterogeneous routing protocols for wireless sensor network. Journal of network and computer applications, 53, 39-56.

Tavares, J., Velez, F., & Ferro, J. (2008). Application of wireless sensor networks to automobiles. Measurement Science Review, 8(3), 65-70.

Tong, M., & Tang, M. (2010, September). LEACH-B: an improved LEACH protocol for wireless sensor network. In 2010 6th international conference on wireless communications networking and mobile computing (WiCOM) (pp. 1-4). IEEE.

Wang, L., Zhang, R., & Geng, S. (2009, September). An energy-balanced ant-based routing protocol for wireless sensor networks. In 2009 5th International Conference on Wireless Communications, Networking and Mobile Computing (pp. 1-4). IEEE.

Wang, G., Wang, Y., & Tao, X. (2009, October). An ant colony clustering routing algorithm for wireless sensor networks. In 2009 Third International Conference on Genetic and Evolutionary Computing (pp. 670-673). IEEE.

Winston, P. H., & Shellard, S. A. (1990). Artificial intelligence at MIT: expanding frontiers. MIT Press.

Xu, Y., Heidemann, J., & Estrin, D. (2001, July). Geography-informed energy conservation for ad hoc routing. In Proceedings of the 7th annual international conference on Mobile computing and networking (pp. 70-84). ACM.

Yang, J., Xu, M., Zhao, W., & Xu, B. (2010). A multipath routing protocol based on clustering and ant colony optimization for wireless sensor networks. Sensors, 10(5), 4521-4540.

Yanagihara, K., Taketsugu, J., Fukui, K., Fukunaga, S., Hara, S., & Kitayama, K. I. (2007). EACLE: Energy-aware clustering scheme with transmission power control for sensor networks. Wireless Personal Communications, 40(3), 401-415.

Yao, X., & Liu, Y. (1997, April). Fast evolution strategies. In International Conference on Evolutionary Programming (pp. 149-161). Springer, Berlin, Heidelberg.

Younis, O., & Fahmy, S. (2004). HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Transactions on mobile computing, (4), 366-379.

Yu, Y., Govindan, R., & Estrin, D. (2001). Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks.

BIOGRAPHY

Amir Seyyedabbasi is a Ph.D. student in the Department of Computer Science and Engineering at Istanbul Sabahattin Zaim University. He received his B.Sc. and M.Sc. degrees in Software Engineering in 2012 and 2015. His research interests include wireless sensor networks, routing protocol and IoT.